Creating Custom Rules with Rule Options

Matt Glass
CISSP, CEH
www.linkedin.com/in/matthewglass2

Overview

Payload detection with content options

Non-payload detection rule options

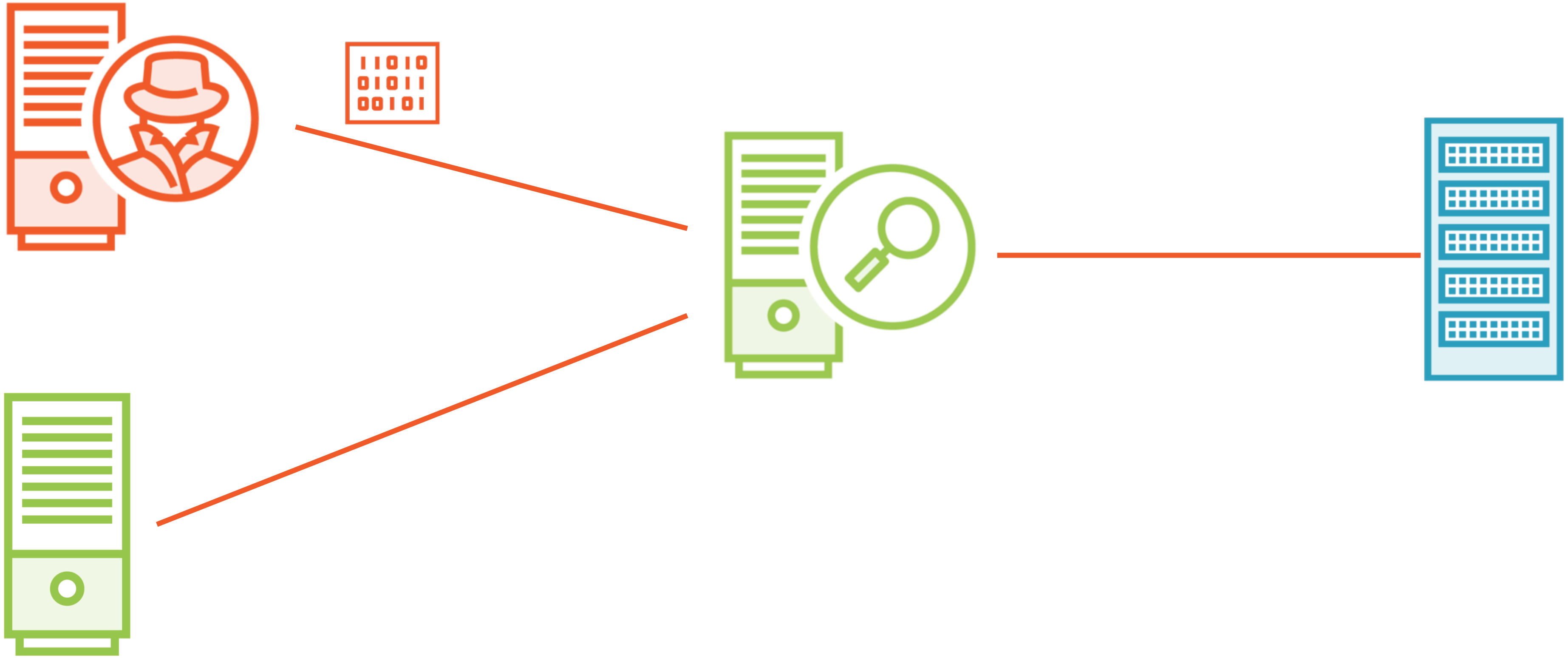
Post-detection logging and tagging

Demos of each of these categories

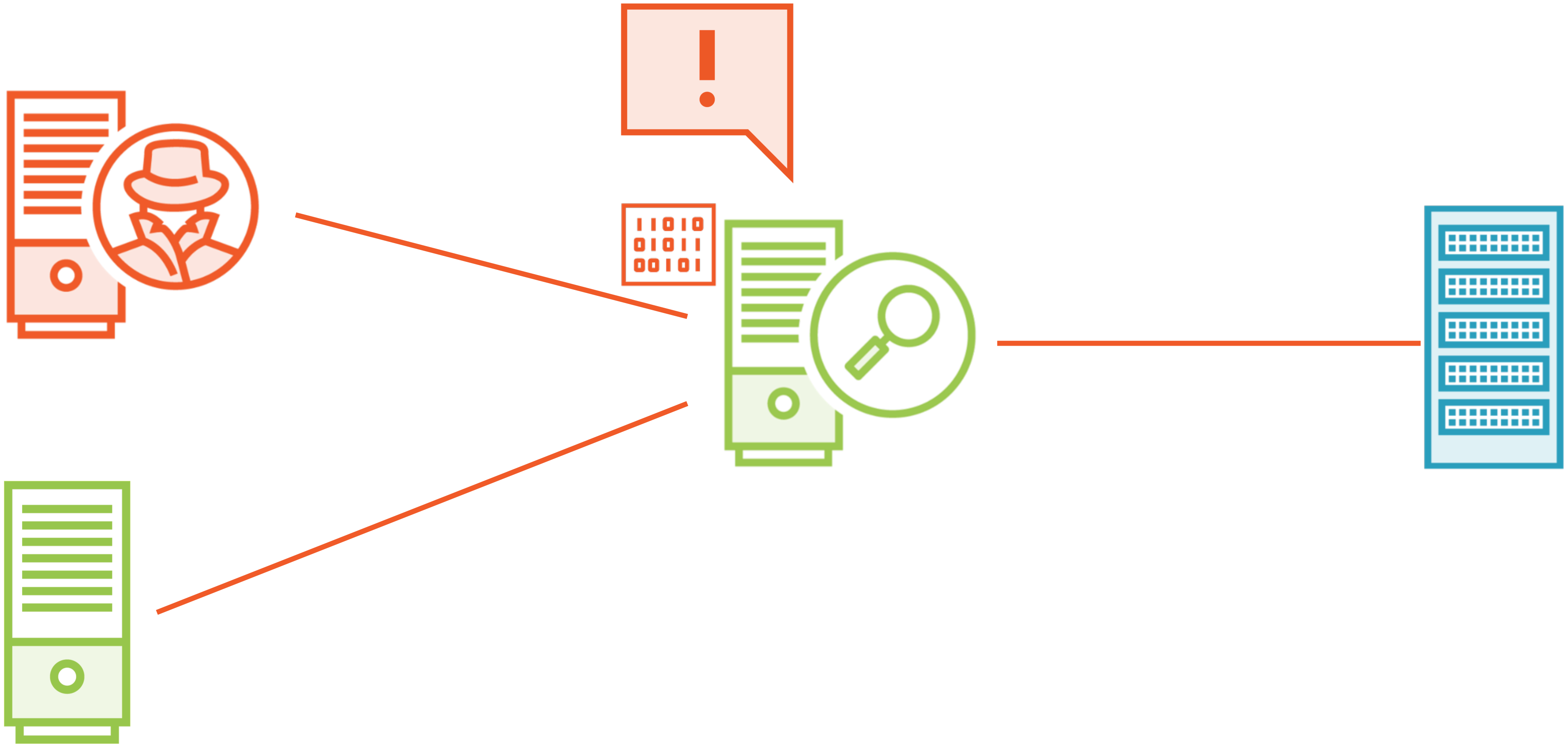
Testing custom rules with target traffic

Payload Detection Rule Options

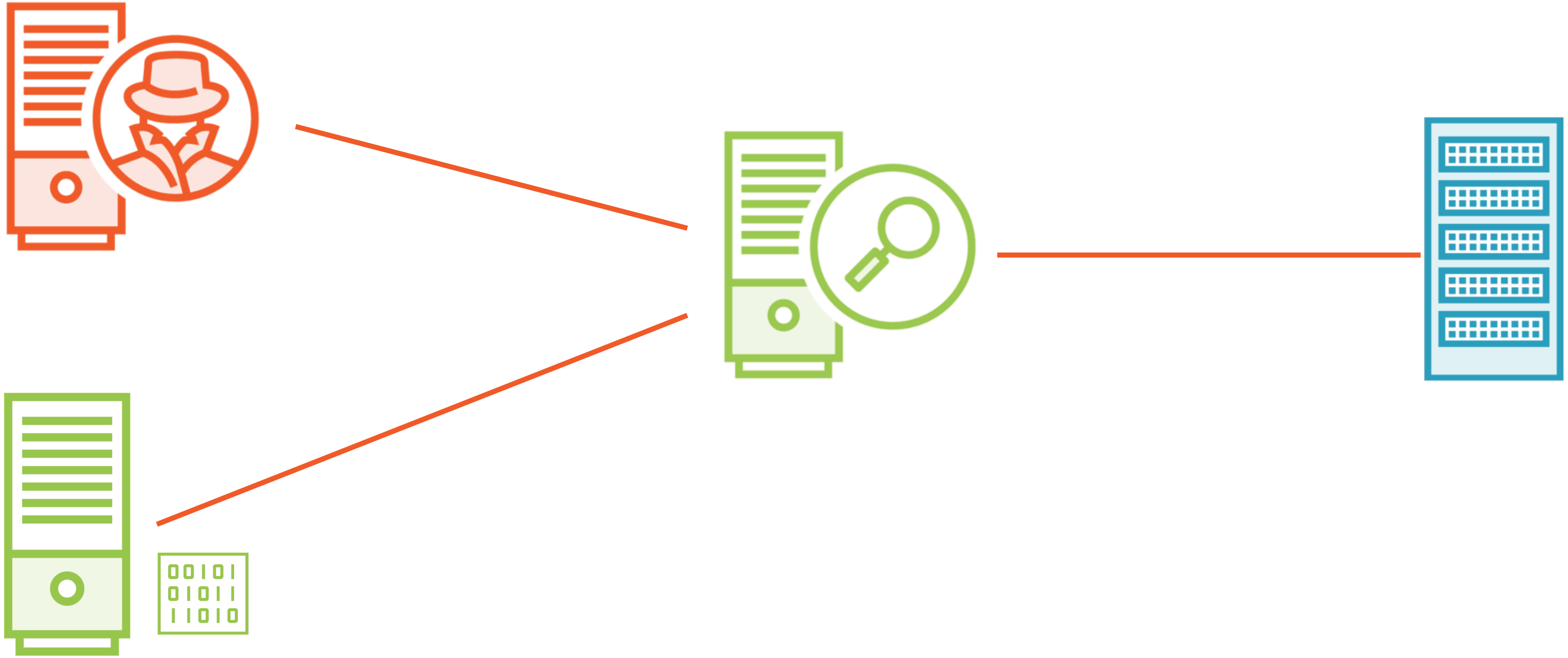
Payload Detection with Content



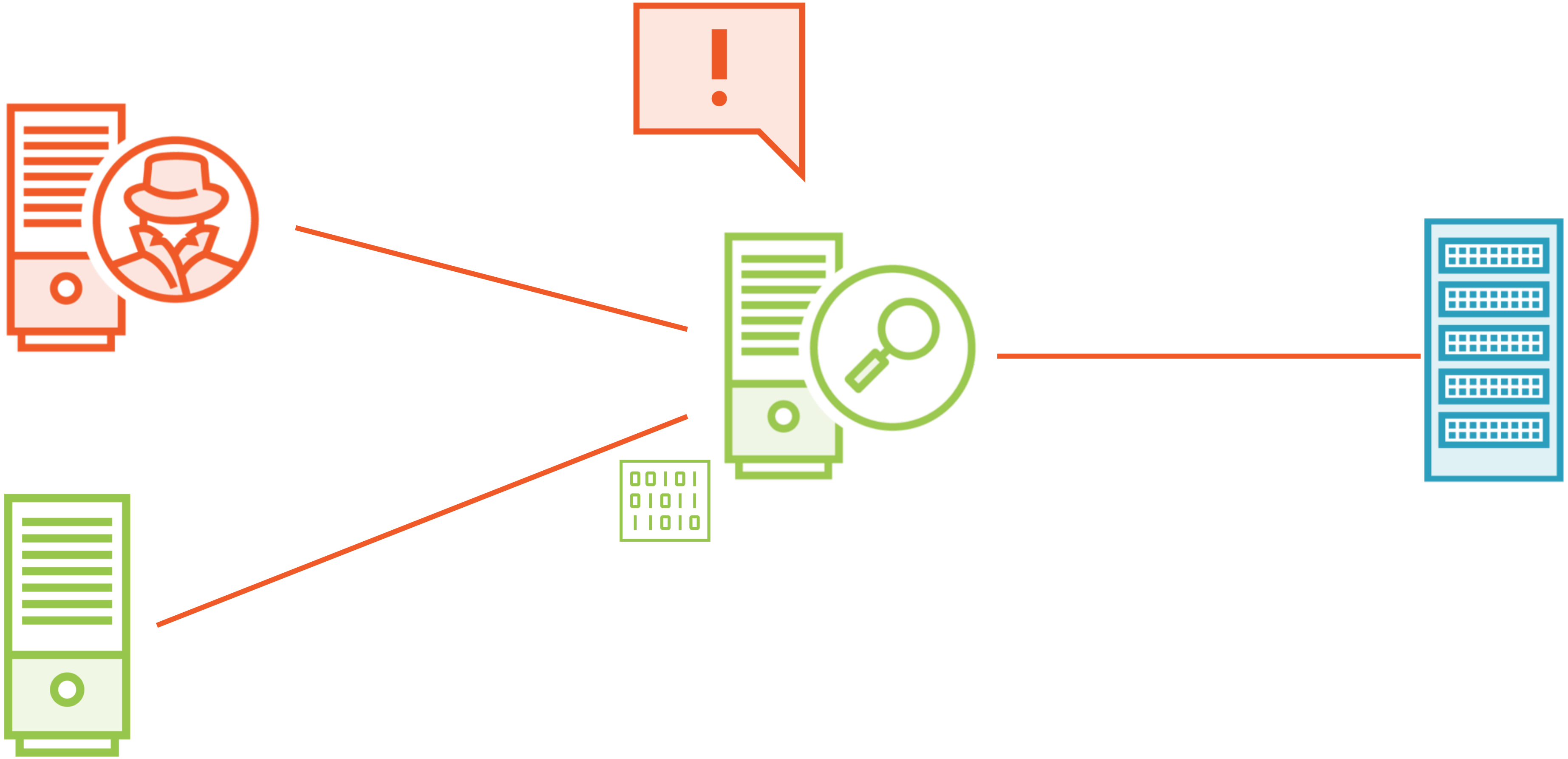
Payload Detection with Content



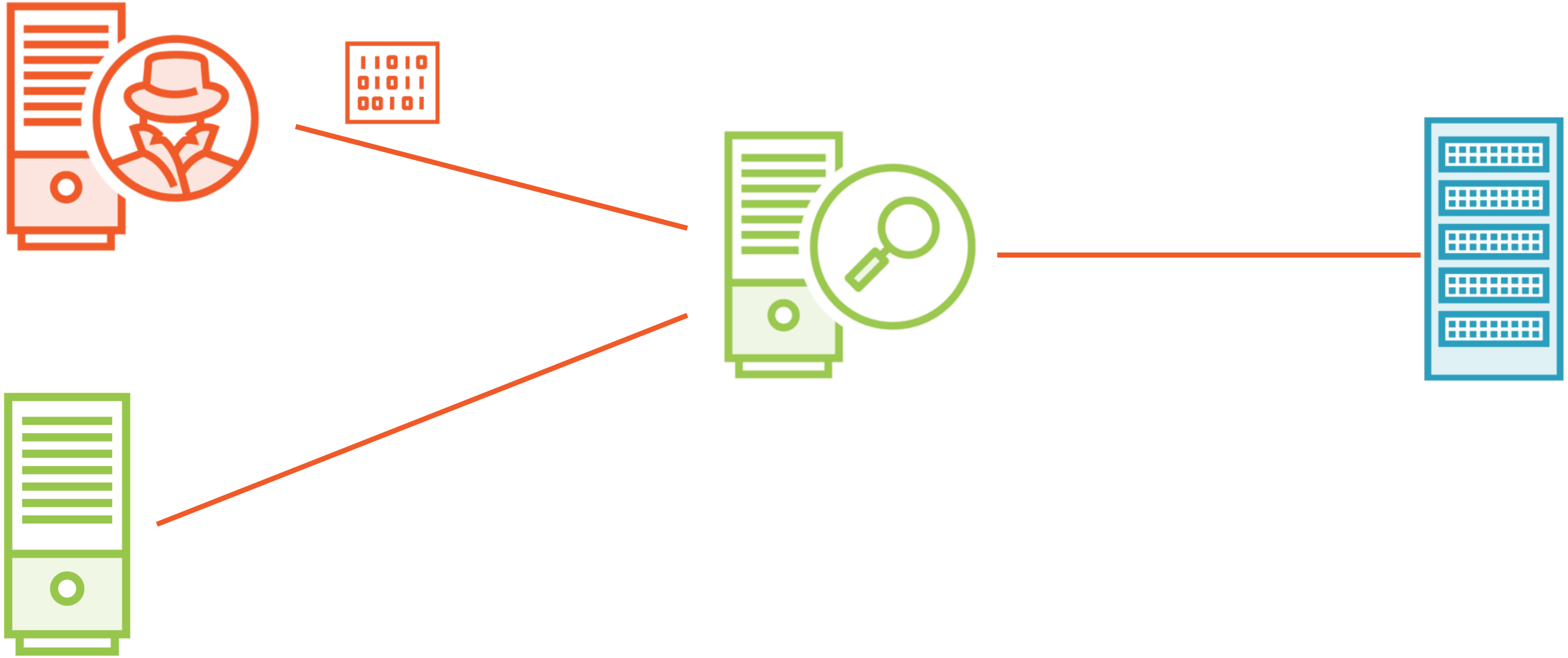
Payload Detection with Content



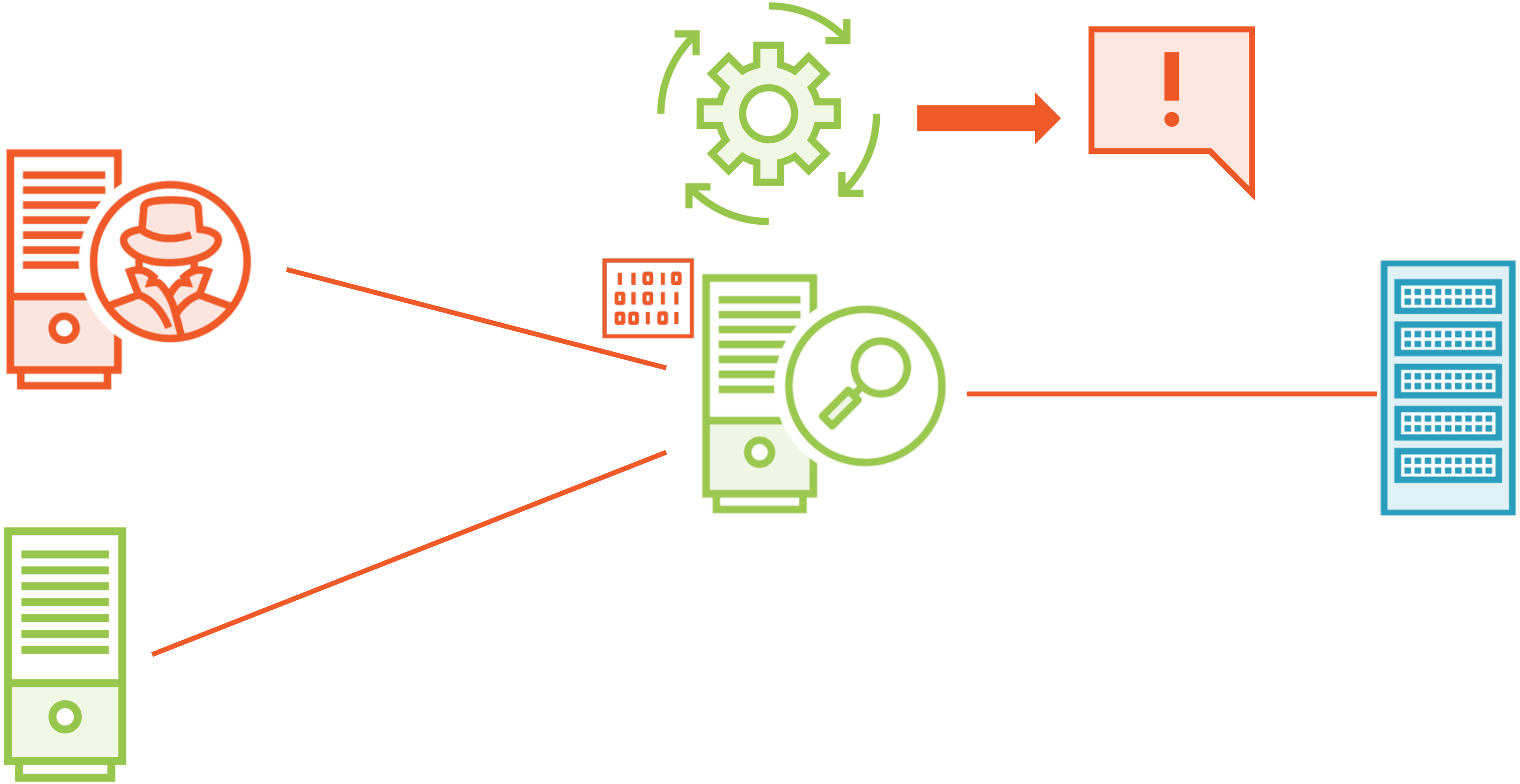
Payload Detection with Content



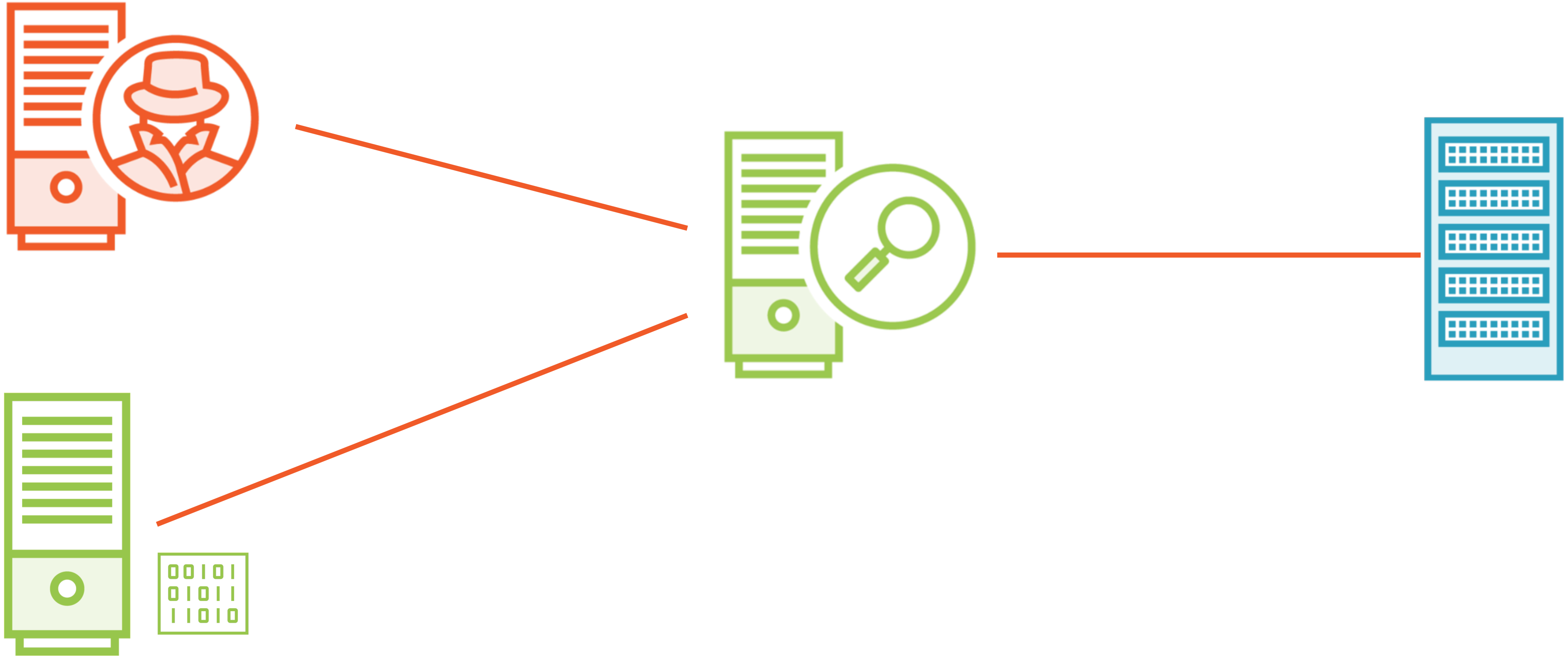
Payload Detection with Content



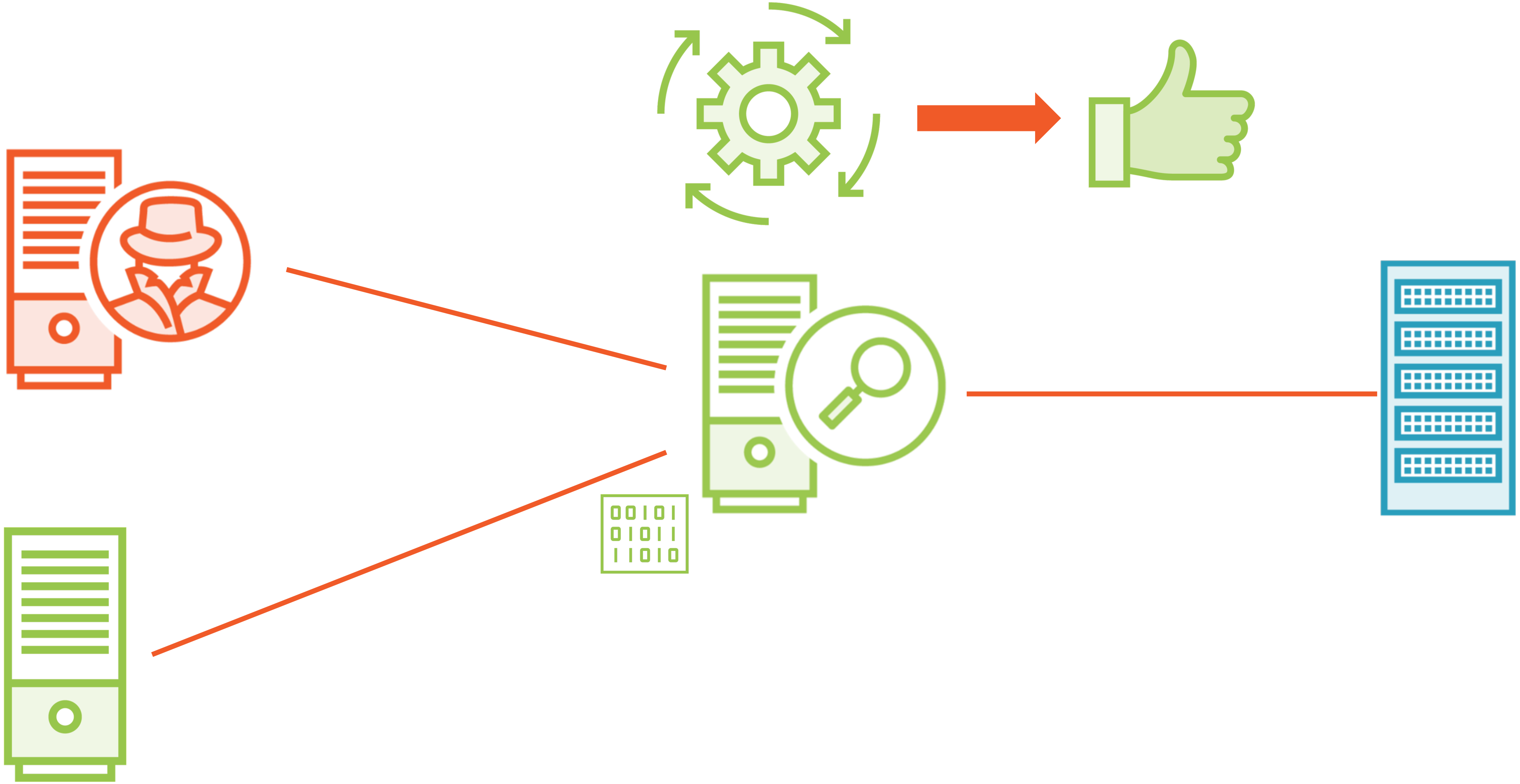
Payload Detection with Content



Payload Detection with Content



Payload Detection with Content



Types of Content Detection

The content option has multiple types and numerous modifiers making it a very flexible method of detecting potential threats.

content

Detection is based on matching payload content to a string

protected_content

Detection is based on matching to a hash value which hides the content

rawbytes

Detection is based on a string of hexedecimal characters

```
alert tcp $EXTERNAL_NET any $HOME_NET 21 (msg:”FTP exploit attempted.”;
protected_content:”
54d626e08c1c802b305dad30b7e54a82f102390cc92c7d4db112048935236e9c”; hash:sha256;
sid:1000001; rev:1;)

alert tcp $EXTERNAL_NET any $HOME_NET 21 (msg:”FTP exploit attempted.”;
content:”|3A 29|”; rawbytes; sid:1000002; rev:1;)
```

Example Protected_Content and Rawbytes Rules

The first rule will generate an alert if a string with the matching SHA256 rule is detected in the packet payload.

The second rule will generate an alert if a string matching the hex code 3A 29 is detected in the packet payload.

Demo

Leveraging the content rule option to protect against an exploit

Address these security goals:

- Reject traffic attempting to exploit the vsftpd backdoor
- Limit the impact on legitimate use of the FTP service
- Verify that this change prevents the backdoor from being executed by testing the exploit

Non-Payload Detection Rule Options

Non-Payload Detection Rule Options

ttl: detects TTL values or ranges between 0 and 255

fragbits: detects if packets are fragmented using the fragmentation bits

dsize: detects packets that are larger than expected

flags: detects certain TCP flags

flow: enables rules based on traffic flow within the same network

```
alert tcp $HOME_NET any $HOME_NET 21 (msg:”FTP exploit attempted.”; flow:to_server;
content:”|3A 29|”; rawbytes; sid:1000002; rev:1;)
```

Example: Using The Flow Option

This rule modifies our previous FTP backdoor rule to only alert when this content is observed flowing to the server from the FTP client.

Demo

Leverage non-payload detection options

- Alert on traceroute attempts
- Drop ICMP packets over 1 KB in size
- Alert on attempt Nmap Xmas scans

Test your rules

Post-Detection Rule Options

Post-Detection Rule Options

detection_filter: sets a rate limit before a rule is triggered

session: logs session data when a rule is triggered

tag: capture additional traffic based on host or session

```
alert tcp $EXTERNAL_NET any $HOME_NET 21 (msg:”vsftp backdoor exploit attempted.”;
content:“:)”; session:printable; sid:1000002; rev:1;)


alert udp $EXTERNAL_NET any $HOME_NET any (msg:”Tracerout detected; ttl:<3;
tag:host,600,seconds,src; classtype:network-scan; sid:1000005; rev:1;)
```

Example: Implementing Session and Tag

Displays the telnet session data after the rule is triggered to determine if this was an attempted exploit.

Captures session traffic from the source IP address that triggered the rule for the next 10 minutes to detect follow on actions.

Demo

Implement post-detection options

- Reject FTP login attempts to 10.0.0.100 from a specific source if brute force attempt is detected. Threshold is 4 logins within 60 seconds.

Test your rule

Summary

Summary

Used content to block a specific attack

Leveraged ttl, dsize, and flags options

Detected traceroute and Xmas scan

Implemented a detection filter

Blocked a simulated brute force attack