Network Security Devices

Network Security Devices

- **Firewall**
  - Analyzes **packet headers** and accepts/rejects **packets** based on policy (rules)
- **Intrusion Detection System (IDS)**
  Intrusion Protection System (IPS)
  - Analyzes packets and connections (including payload)
    - IDS – Logs packets matching signatures
    - IPS – Blocks packets matching signatures
- **Honeypots**
  - Lure to invite attack and provide early warning

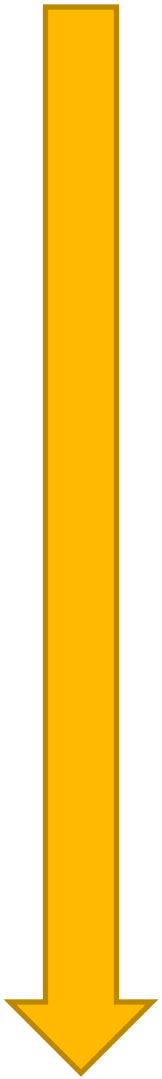
Firewalls

Firewall Basics

- Firewall makes decisions about whether to permit or reject *individual packets* based on **rules**
- Rule can consider
  - Ingress port? Egress port?
  - Source address? Source port?
  - Destination address? Destination port?
  - Protocol? (TCP, UDP, …)
  - Connection state
    - **Stateful firewalls** can track communication *across* related packets

Firewall Rules

- Security recommendation: **Default Deny**
  - Only specific traffic is permitted on the network – everything else is denied (by default)
  - In contrast to Default Accept, where everything is permitted unless explicitly blocked
- Rules are **sorted** and examined from top to bottom
  - **The first rule matching a packet wins! – all subsequent rules are ignored**



Each incoming packet is matched against rules from

TOP

to

BOTTOM

After first match, subsequent rules are ignored

https://www.sans.org/media/score/checklists/FirewallChecklist.pdf

Most important rule is LAST! **Default Deny**

- From any source (IP/port)
- To any destination (IP/port)
- For any protocol
- DROP IT ALL
- *And optionally log to see what packets reach this far into the firewall rules*

Deny all and log

https://www.sans.org/media/score/checklists/FirewallChecklist.pdf

Rules that explicitly deny and alert on known-suspicious traffic

May subscribe to vendor that updates these rules based on current threat patterns

Deny and alert

- Alert IT about suspicious traffic

Deny all and log

https://www.sans.org/media/score/checklists/FirewallChecklist.pdf

What IS supposed to be allowed on your network?

Rules permitting both user traffic and IT management traffic

Rules permitting user traffic

- Permit HTTP/HTTPS to webserver
- …

Rules permitting IT traffic

- Permit SNMP traffic to network monitoring server
- …

Deny and alert

- Alert IT about suspicious traffic

Deny all and log

https://www.sans.org/media/score/checklists/FirewallChecklist.pdf

Rules blocking spoofing

- Block private addresses
- Block internal LAN addresses coming from WAN)

Rules permitting user traffic

- Permit HTTP/HTTPS to webserver
- …

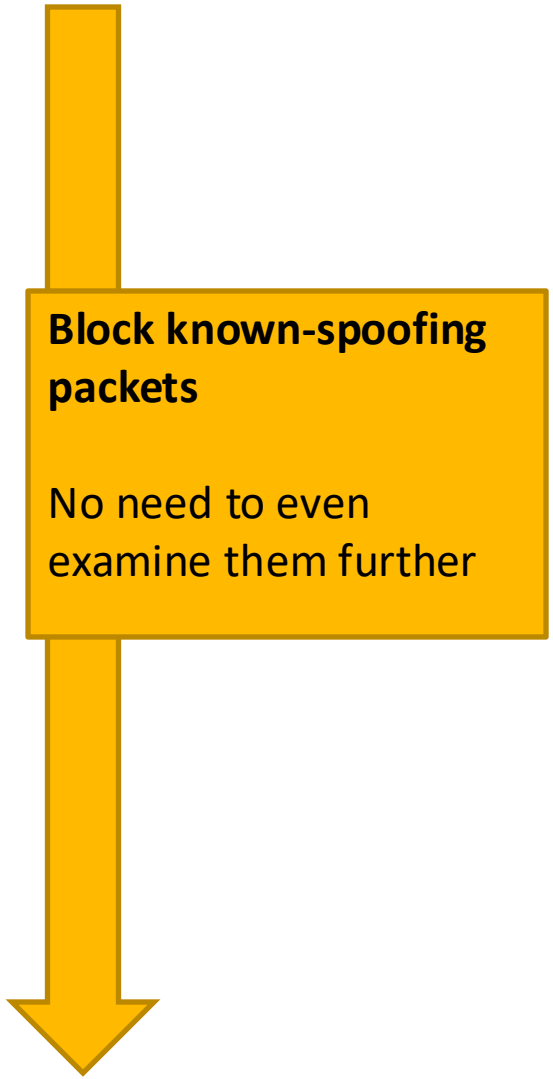
Rules permitting IT traffic

- Permit SNMP traffic to network monitoring server
- …

Deny and alert

- Alert IT about suspicious traffic

Deny all and log



https://www.sans.org/media/score/checklists/FirewallChecklist.pdf

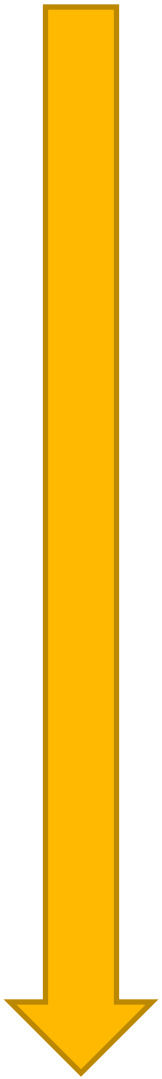
Rules blocking spoofing
- Block private addresses
- Block internal LAN addresses coming from WAN)

Rules permitting user traffic
- Permit HTTP/HTTPS to webserver
- …

Rules permitting IT traffic
- Permit SNMP traffic to network monitoring server
- …

Deny and alert
- Alert IT about suspicious traffic

Deny all and log



Each incoming packet is matched against rules from

TOP

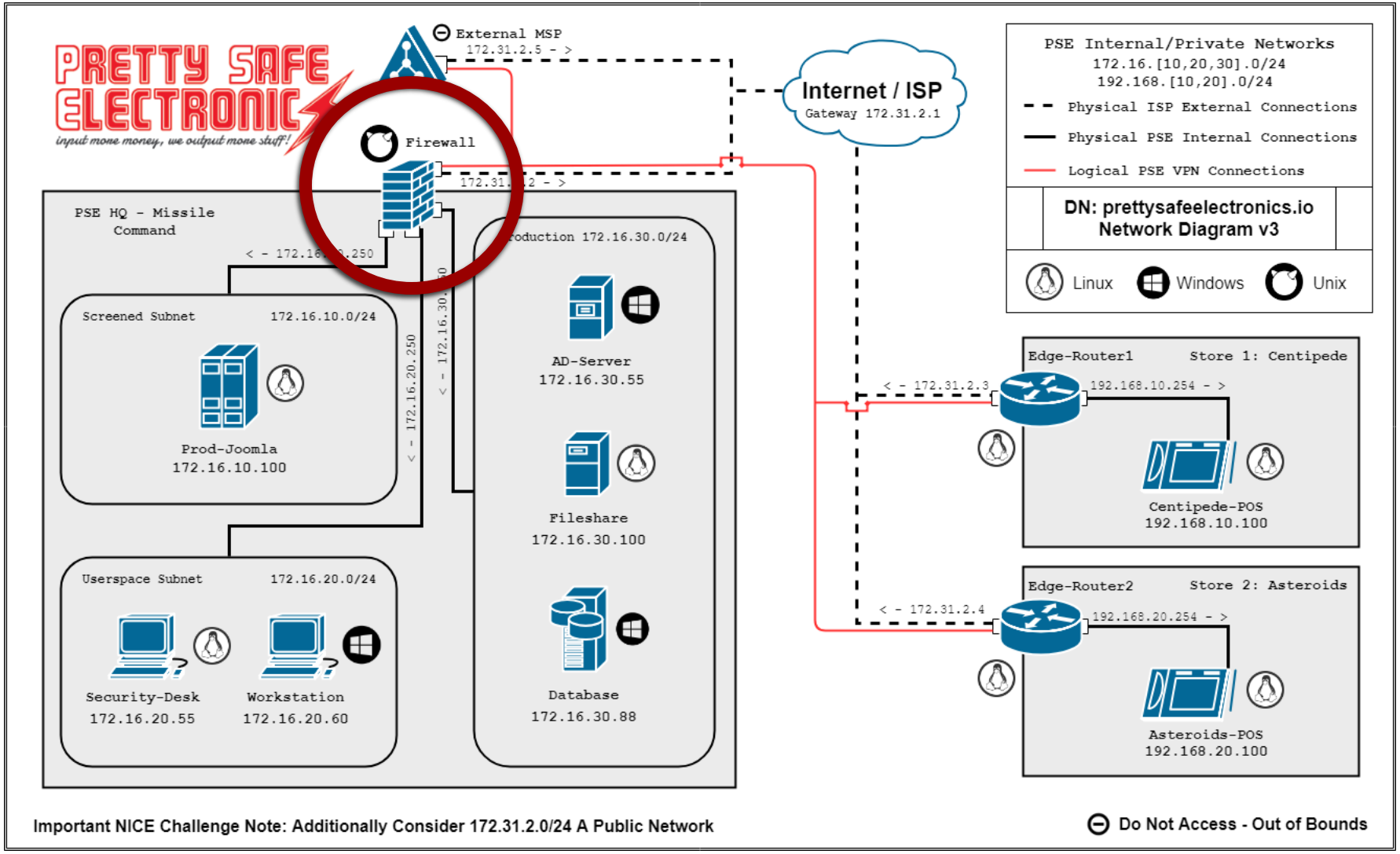
to

BOTTOM

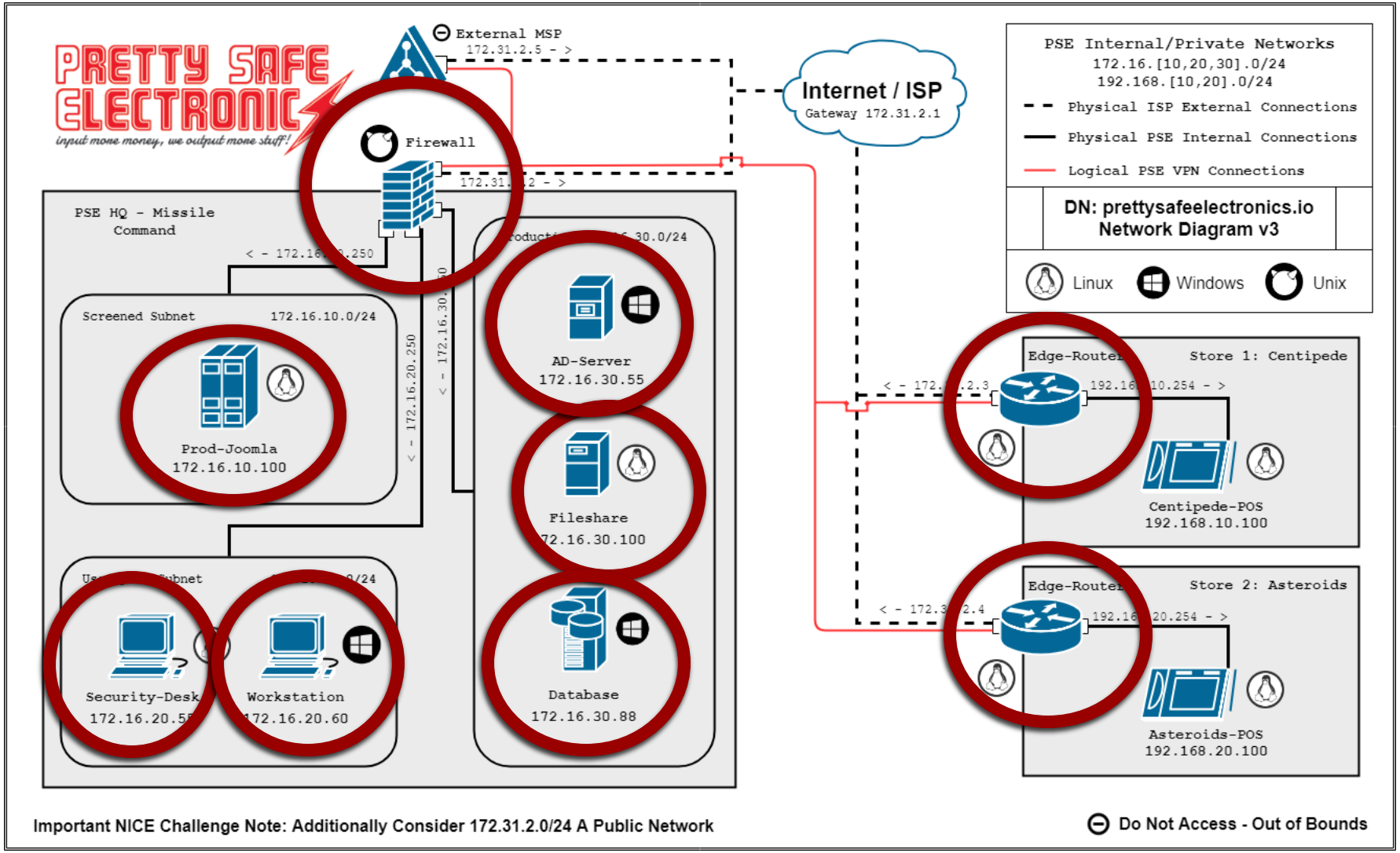
After first match, subsequent rules are ignored

https://www.sans.org/media/score/checklists/FirewallChecklist.pdf

Perimeter Defense – Network Firewall Only



Defense-In-Depth – Network and Host-Based Firewalls



Intrusion Detection Systems (IDS)

Intrusion Detection System (IDS)

- Host-based
  - Installed on local machine
  - Monitor local programs, logs, security settings
- Network-based
  - Monitors network activity via deep packet inspection
  - Signature-based
    - Compares packets with known signatures
    - Examples: Snort, Zeek (*formerly Bro*), Suricata
  - Anomaly-based
    - Learns normal behavior of network and alerts on differences

https://snort.org/resources
https://www.youtube.com/watch?v=W1pb9DFCXLw

Intro to Snort

https://snort.org/resources
https://www.youtube.com/watch?v=8T8XVoNqMbc

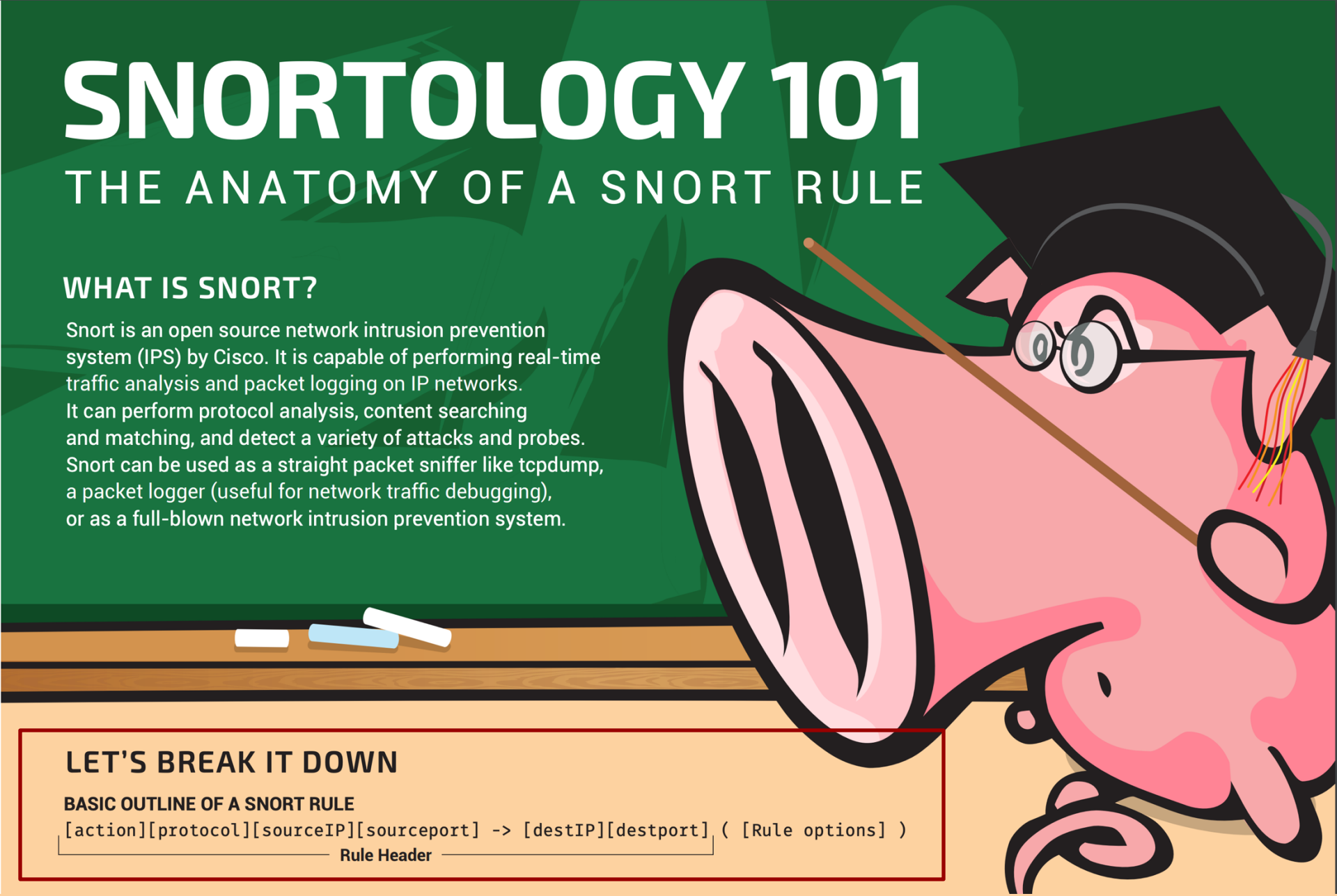
Rule Writing

SNORTOLOGY 101

THE ANATOMY OF A SNORT RULE

WHAT IS SNORT?

Snort is an open source network intrusion prevention system (IPS) by Cisco. It is capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching and matching, and detect a variety of attacks and probes. Snort can be used as a straight packet sniffer like tcpdump, a packet logger (useful for network traffic debugging), or as a full-blown network intrusion prevention system.



LET'S BREAK IT DOWN

BASIC OUTLINE OF A SNORT RULE

```
[action][protocol][sourceIP][sourceport] -> [destIP][destport] ( [Rule options] )
└──────────────────────────── Rule Header ──────────────────────────┘
```

RULE HEADER

The rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information.

alert **Action to take (option)** The first item in a rule is the rule action. The rule action tells Snort what to do when it finds a packet that matches the rule criteria (usually alert).

tcp **Type of traffic (protocol)** The next field in a rule is the protocol. There are four protocols that Snort currently analyzes for suspicious behavior - TCP, UDP, ICMP, and IP.

$EXTERNAL_NET **Source address(es)** variable or literal

$HTTP_PORTS **Source port(s)** variable or literal

-> **Direction operator** The direction operator -> indicates the orientation of the traffic to which the rule applies.

$HOME_NET **Destination address(es)** variable or literal

any **Destination port(s)** variable or literal

EXAMPLE

Rule Header

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any
```

Message

```
msg: "BROWSER-IE Microsoft Internet Explorer
CacheSize exploit attempt";
```

Flow

```
flow: to_client,established;
```

Detection

```
file_data;
        content:"recordset"; offset:14; depth:9;
        content:".CacheSize"; distance:0; within:100;
        pcre:"/CacheSize\s*=\s*/";
        byte_test:10,>,0x3ffffffe,0,relative,string;
```

Metadata

```
policy max-detect-ips drop, service http;
```

References

```
reference:cve,2016-8077;
```

Classification

```
classtype: attempted-user;
```

Signature ID

```
sid:65535;rev:1;
```

RULE OPTIONS

Rule options form the heart of Snort's intrusion detection engine combining ease of use with power and flexibility. All Snort rule options are separated from each other using a semicolon (;). Rule option keywords are separated from their arguments with a colon (:).

GENERAL RULE OPTIONS

Message A meaningful message typically includes what the rule is detecting. The msg rule option tells Snort what to output when the rule matches. It is a simple text string.

Flow For the rule to fire, specifies which direction the network traffic is going. The flow keyword is used in conjunction with TCP stream reassembly. It allows rules to only apply to certain directions of the traffic flow.

Reference The reference keyword allows rules to include references to external sources of information.

Classtype The classtype keyword is how Snort shares what the effect of a successful attack would be.

sid/rev The snort id is a unique identifier for each rule. This information allows output plugins to identify rules easily and should be used with the rev (revision) keyword.

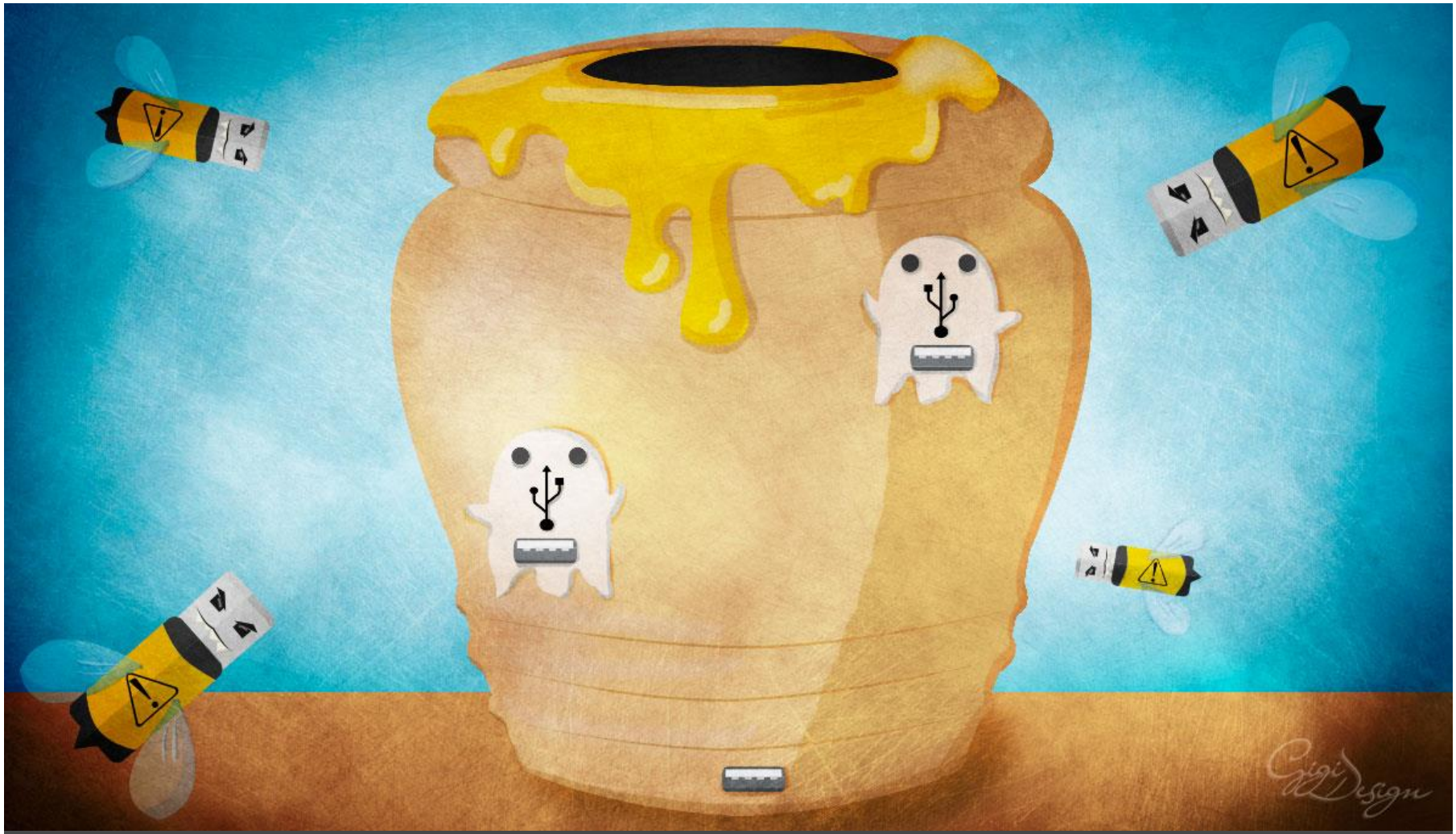
DETECTION OPTIONS

Content This important feature allows the user to set rules that search for specific content in the packet payload and trigger response based on that data. The option data can contain mixed text and binary data.

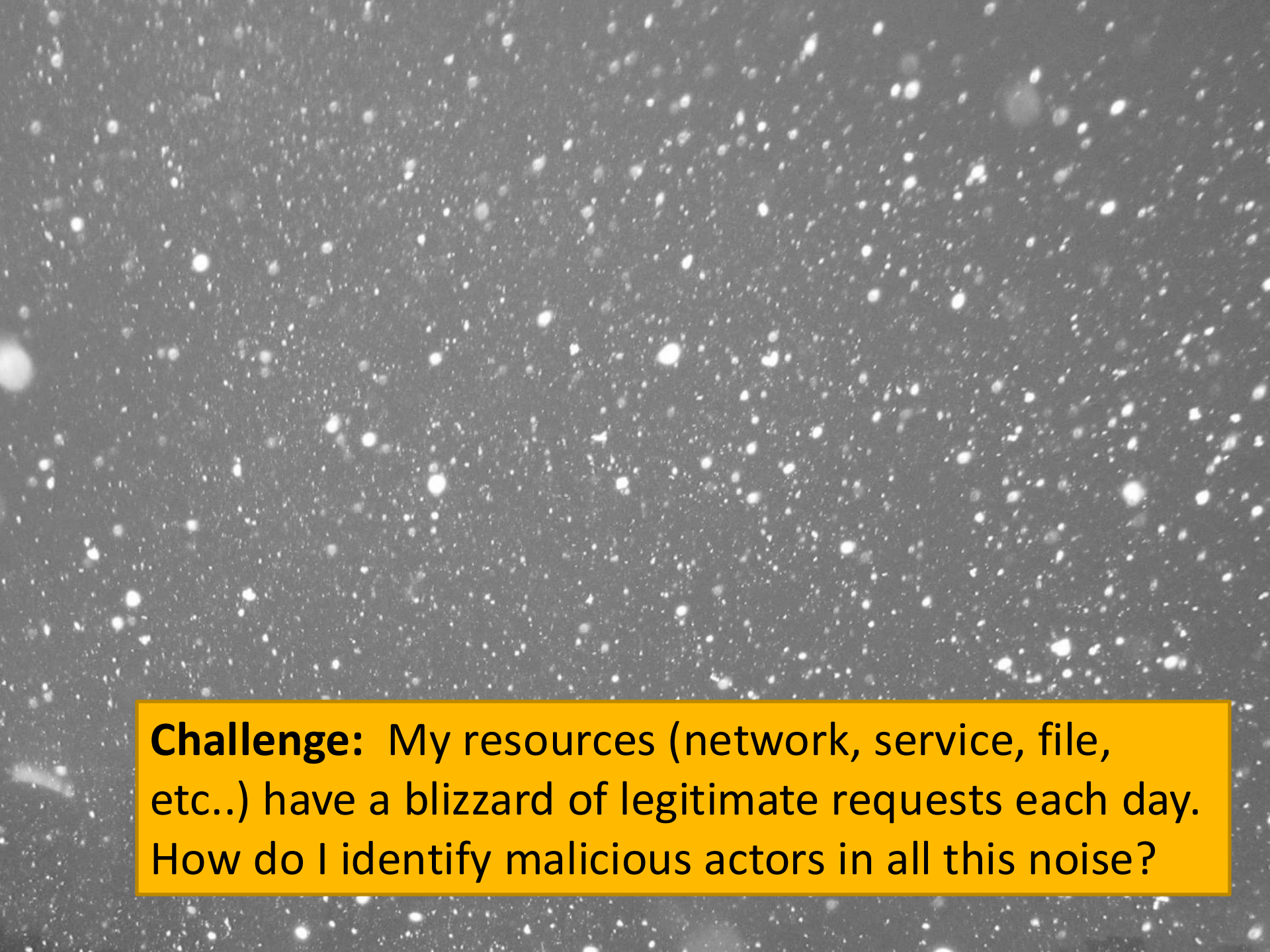
- **distance/offset** These keywords allow the rule writer to specify where to start searching relative to the beginning of the payload or the beginning of a content match.
- **within/depth** These keywords allow the rule write to specify how far forward to search relative to the end of a previous content match and, once that content match is found, how far to search for it.

PCRE The pcre keyword allows rules to be written using perl compatible regular expressions which allows for more complex matches than simple content matches.

Byte test The byte_test options allows a rule to test a number of bytes against a specific value in binary.



Honeypots


Challenge: My resources (network, service, file, etc..) have a blizzard of legitimate requests each day. How do I identify malicious actors in all this noise?

Honeypots

- A resource that has *no value to legitimate users* but is attractive to attackers
  - Greatly simplifies alerting, as activity on resource is almost always malicious
- **Alert** – Provide early warning of attack (rather than FBI notification 6+ months later)
- **Lure –** Make the attackers waste lots of time here
- **Monitor** – What are the attackers trying to do?
  - Commands entered?
  - Malware uploaded?

Honeypot Use Cases

- **Production systems**
  - *Goal: Protect our current systems*
  - Alert to ongoing attacks that are missed by pattern-based IDS
  - Deterrence (potentially?) if attackers realize they are being monitored
  - Useful for all businesses

Honeypot Use Cases

- **Research**
  - *Goal: Study attackers*
  - Learn about attacker skill level, tools, motives, origin, …
  - Useful for academics, governments, security researchers, …

Honeypot Interactivity

What kind of interaction can the attacker have with the honeypot?

(in comparison to a *real* vulnerable system)

Low interactivity

Medium interactivity

High interactivity

Low Interaction Honeypot

- Minimal functionality
  - Example: Listen on all TCP ports, accept all connections, and receive data for up to 20 seconds. Send minimal or no replies.
- Pros: Minimal danger to other systems, simple implementation
- Cons: Minimal information learned
  - Source IP, source port, payload sent (if any)

Medium Interaction Honeypot

- System emulates vulnerabilities only
  - Partial simulation of a real system
  - Attacker can't do much after exploiting vulnerability
- Pros: Reduced danger to other systems
- Cons: Some information learned
  - Attacker is present (source IP)
  - Attacker used specific vulnerability to gain entry
  - What would attacker have done once inside?

High Interaction Honeypot

- Attacker can interact with system at all levels
  - Probe, attack, and compromise
  - Pivot through system for additional attacks
- Equivalent to a real system with hidden monitoring infrastructure
  - Key logging, network logging, file logging, …
  - Data control – Limit where the attacker can go *after* entering honeypot
- Pros: High level of information learned
  - Where are the attackers coming from?
  - What is their skill level?
  - What tools are they using?
- Cons: Risk in letting attacker own our system?
  - Attack the rest of our network?
  - Attack systems outside our organization?
  - Store/distribute illegal content?

Honeypots

- Wide range of possible implementations
  - Dedicated machine
  - Virtual machine
  - Special service on a host
  - Special file on a host
- Never meant for legitimate use
  - Any access is either accidental *or* malicious

Quick and Dirty?

- **Q:** Why don't I just install some old unpatched OS and service software in my datacenter? It'll be attractive to attackers, right?
- **A:** Method would be possible if the *only* system on your network was the honeypot
  - Risky in a full datacenter
  - What if the attackers springboard from the honeypot system to attack legitimate services next?

Thinkst Canary

- Thinkst Applied Research: South African security company
- **Tripwire honeypot**
- Offer a honeypot service (physical hardware, virtual machine, or AWS)
- Paid product ($5k/year for 2 Canaries)

https://canary.tools/

https://vimeo.com/337352561

Thinkst Canary

- Configurable to many “personalities”
  - Windows Server 2008, 2003, 10, 8, 7, XP, …
  - Diskstation NAS
  - VMWare ESXi
  - Linux
  - OS X
  - Cisco router, Dell switch
  - Rockwell Automation PLC, Siemens Simatic PLC
  - And more?
- They’re “emulating” these devices to a certain level of fidelity – not really running Windows…

Thinkst Canary

- Configurable with interesting services
  - SSH
  - Telnet
  - SMB (Windows file sharing)
  - Web server (usual suspects, JBoss, VMWare management console, Sharepoint, …)
    - Upload your own fake website, including SSL cert
  - Database
- File shares can be full of fake interesting data
  - Payroll.xls

Thinkst Canary

- Alerts when malicious activity detected
  - SMS, Emails, Slack
  - Visible on external dashboard

Cowrie Honeypot

- **SSH / SFTP honeypot**
  - Fake filesystem (resembles Debian 5.0) with ability to add/remove files
  - Potential (?) for fake file contents, e.g. /etc/passwd
  - SFTP and SCP file uploads/downloads
  - SSH exec commands (`ssh user@host 'cat /etc/passwd'`)
  - SSH tunneling / SSH proxy logging
  - Integration with ELK (ElasticSearch, Logstash, Kibana)

https://github.com/cowrie/cowrie

https://www.cowrie.org/

Dionaea Honeypot

Venus flytrap (*Dionaea muscipula)*

- **Malware trap honeypot**
- Identify attackers trying to exploit network server vulnerabilities and capture a copy of the malware they are attempting to run
- Emulates variety of network protocols that attackers are interested in (including vulnerabilities!)
  - FTP, HTTP, Memcache, MySQL, MSSQL, pptp, sib, SMB, …

https://github.com/DinoTools/dionaea

https://www.javelin-networks.com/honeypotbuster

https://github.com/JavelinNetworks/HoneypotBuster

Detectability

- Attackers can obtain the same honeypot software as defenders, and write / distribute fingerprinting scripts to avoid them
- *Constant cat and mouse game*

Other Resources

- "Awesome Honeypots"
  - https://github.com/paralax/awesome-honeypots
  - Curated list
  - *More honeypots (and associated tools) than you ever knew about!*