Written Assignment Unit 3

MICS-252, Fall 2024

Prepared by: Karl-Johan Westhoff
email: kjwesthoff@berkeley.edu
UC Berkleley School of Information
MICS Course 252 Fall 2024 (Kristy Westphal)

1 Introduction

For the first 2 weeks we have been working on web applications, I decided to continue on this topic for the SOC Tools, as discussed in class a SOC processes a firehose of information, and deploys fascinating tools[1] and processes to do that. I wanted to look at some of the tools that collect and generate the data:

- Web Application Firewalls (WAF), which apart from blocking stuff, 'of course' also logs its findings
- Intrusion Detection Systems (IDS), which monitors network traffic and reports suspicious activity

1.1 'Things' going to the cloud

Cloud web applications are often distributed into "microservice" architectures where data is retrieved from APIs serving multiple applications rather than managed and retrieved directly from e.g. a dedicated database[2].

The distributed-ness poses a new set of challenges for web security. The distributed architecture allows for a lot of different technologies to work together, often developed using "someone else's code" from open source libraries. Free Open Source Software (FOSS) is wonderful when you are a developer: you can usually find that someone has already solved the problem you are working on. But, there may be un-discovered vulnerabilities in the code, the risk needs to be mitigated somehow by:

- Keeping inventory of the code and versions used (SBOM).
- Deploy systems for detecting and preventing attacks (WAF and IDS).

With modern DevOps deployment where API's and services can be developed and deployed quickly. Security Operations (SecOps) needed to be adapted and security was shoehorned into the DevOps acronym, it is now referred to as DevSecOps[3]. The intention being that security is not to be regarded as an

---

[1] Security information and event management (SIEM)

[2] I am myself 'guilty' of developing an application with more microservices than users at one point..

[3] Shannon Lietz of Intuit is often credited with coining the term[1]

afterthought but part of the development cycles.

With distributed applications there is a larger attack surface, there are more http endpoints that can be hit and the OWASP top 10 applies to all of them. The A01:2021-Broken Access control [2] has probably moved to 1st. place in 2021 as Access and Identity control needs to be managed for all http endpoints in the distributed applications.

2 Web Application Firewalls

WAF's protect web applications and covers many of the items in the OWASP top 10 list [2]. WAF's scope is limited to the HTTP/S Protocol and WAF's are focused on analysis of http requests and responses. WAF's can be deployed in front of web applications or as part of other infrastructure e.g. reverse proxies and cloud load balancers.

2.1 WAF vs. "Ordinary" Firewall

Firewalls in general works by allowing or blocking traffic on specific transport layer ports and ranges of IP addresses. Sometimes firewalls also work as 'stateful' where in and out going package ip's and ports are compared and grouped into sessions (i.e related traffic between the server and a client) and evaluated based on the relations between packets. More fancy firewalls also 'open' the packets and inspects parts of the packet payload, this requires more computing resources.

I guess a WAF falls into the 'fancy firewall' category: HTTP/S traffic happens at the application layer of the OSI model and hence relates to the TCP/UDP packet payload. However, a difference to ordinary firewalls is that WAF's are specialized to work on http requests, since the traffic needs to be decrypted and parsed by the web server anyway, some the resource overhead from the WAF can be mitigated by doing some of the web servers work on the WAF while checking the traffic.

2.2 WAF in the cloud

Much like with ordinary firewalls where one instance protects a whole subnet, WAF's can be deployed with a reverse proxy in front of a distributed web application, putting a WAF in front of the traffic can help mitigate some of the risk associated with FOSS and the complexity associated with distributed web apps.

2.3 WAF examples

As WAF's are based on analyzing the packet payloads and http query strings; 'it runs some code', checking if attacks are occurring e.g. by looking for:

- 'special' characters for injection attacks
- if passphrase spraying is being attempted
- etc. from the OWASP top 10

For that, it needs to keep track of multiple packets to track sessions and TCP streams, many vendors advertize leveraging machine learning for this in their WAF's[3].

Types of WAF's: WAF's are deployed in different settings depending on how the web applications they protect are deployed, in all cases WAF's need adequate processing power and network bandwidth.

- **Host Based**: On the same physical hardware and OS as the web application
- **Network Based**: On a separate 'box' on the network, for example with a reverse proxy
- **Cloud Based**: As part of the cloud provider's monitoring and logging systems, usually provided as a paid-for service[4] by the cloud provider

WAF software providers: Most modern web applications are cloud deployed, A list of ten popular WAF's is presented in [3], some examples are:

- **Cloud vendor based**: Provided as a service where you host you web application:
  - "Azure WAFv2 [4]"
  - "AWS WAF [5]"
- **Security providers**: Managed solutions for outsourcing the concern:
  - "Cloudflare WAF [6]"
  - "AppTrana Manage WAF [7]"
- **Open Source**: For research and budget oriented companies e.g. NGO's, requires some in-house security expertise:
  - "open-appsec [8]"
  - "OWASP Corza [9]"

---

[4] ['*Your Acronym Here*']aaS

3 Intrusion Detection Systems

Intrusion detection systems (IDS) monitor network traffic and reports/alerts if it recognizes suspicious activity, sometimes IDS also actively blocks traffic, then referred to as an Intrusion Prevention System (IPS). IDS systems often deliver data to a SIEM for further analysis. IDS's can be deployed both on individual hosts or at strategic places in networks, for example on a switch's promiscuous/mirror port analyzing all network traffic going through the switch. Just like WAF's IDS's need to analyze network packets, a key difference is that IDS's are not limited to HTTP/S traffic, but they are scoped to the full spread of protocols running on the transport layer.

IDS's usually work by applying rules to simple packet analysis in individual transport layer packets, checking for source/destination IP's, ports and which protocols are transmitted (much like a firewall). IDS also keep track of series of packets and their relations in sessions (stateful protocol analysis). Furthermore, most IDS's also look at anomalies in traffic, many vendors advertise deploying machine learning here. IDS can detect more advanced attacks, one example is the 'EternalBlue' attack on Microsoft's SMB protocol, the attack was detectable by looking at a specific pattern of TCP traffic on port 446, and quickly distributing IDS rules helped mitigate the attack (See Snort Rule example in Appendix A).

IDS applies predefined rules often based on known patterns for malicious traffic (signatures) to trigger alerts. IDS rules need to be maintained and updated to reflect the threats on the internet.

3.1 IDS in the cloud

Network traffic in cloud is virtualized and you cannot just plug an IDS into a mirror port on a switch to monitor the network, you are somewhat reliant on the cloud provider to deliver data for monitoring. IDS could be deployed on every cloud instance but that would create a large workload, both of extra compute resources on each instance, and a high workload for maintaining and deploying signatures to many instances.

Case study: Snort on Oracle Cloud For the MICS Cyber210 networking class we wanted to build an "Attack and Defend" lab with a monitored network containing vulnerable hosts, and detect attacks on these using Snort[10]. Instead of actually using hardware we decided to build a virtual lab using Oracle Cloud[5]. It turned out it wasn't so easy to rig a cloud network with host based software. Apart from separating the network to hide the vulnerable machines from the internet (obviously). We wanted to monitor traffic on a subnet, to simulate the promiscuous/mirror port on a fictive switch, we had to route the traffic from the

---

[5] The intention was to deploy deliberately vulnerable infrastructure to test, not all cloud providers make this easy. We found that Oracle Cloud has a 30day/300$ trial period which we decided to burn on the project.

vulnerable machines to the IDS using a load balancer and virtual tap's (VTAP) on each of the instances we wanted to monitor to a Ubuntu instance running Snort see Figure 1

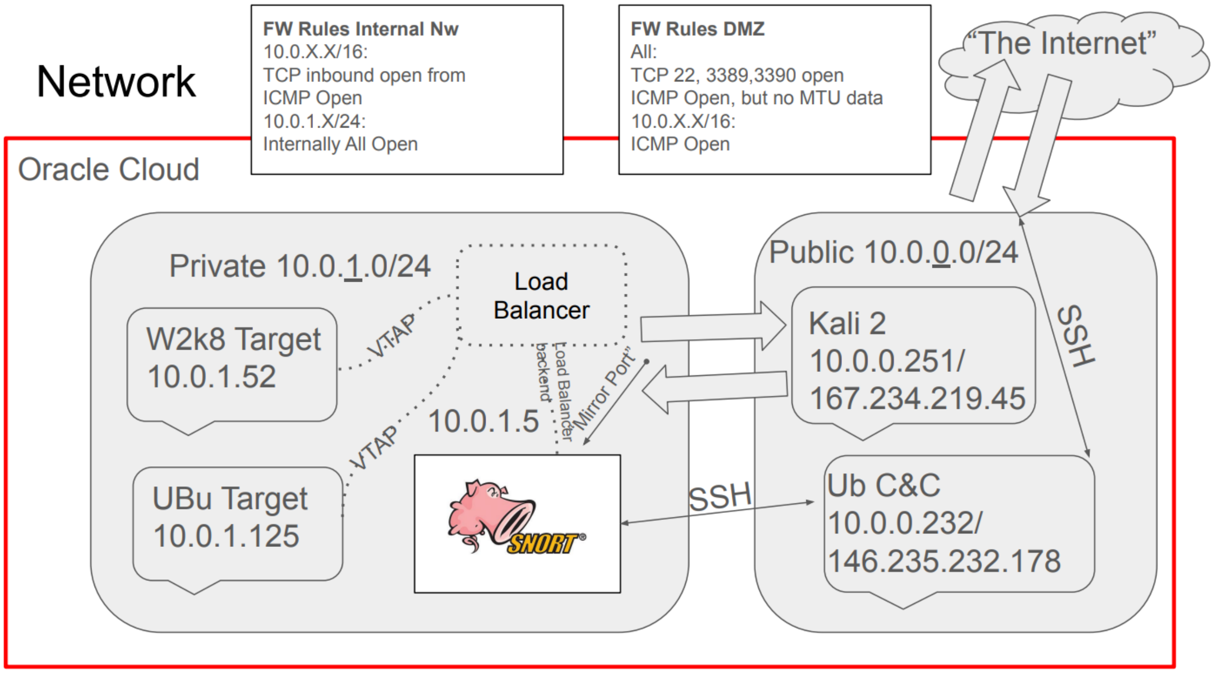


Figure 1: *Attack and Defend Lab Diagram, for a project for Cyber210. The idea is that attacks are performed on the targets from the Kali instance (attacker logged in via RDP), and detected by defenders (subscribing to Snort alerts via SSH). The dotted lines indicate the traffic workarounds simulating a physical mirrored port on a switch using VTAPs and a load balancer.*

3.2 IDS Examples

There seems to be some consensus that IDS's/IPS can be split into 2 categories:

- **Host-based Intrusion Detection System (HIDS)**:
  - Installed on every host, may trace network activity to a specific application on that host.
  - Uses resources from the host and needs to be maintained and signatures updated on that host.
- **Network Intrusion Detection System (NIDS)**:
  - Monitors a whole subnet using one instance (less to maintain)
  - Needs to keep up with the traffic (may be a challenge on large networks)

Beyond **'H'** and **'N'** IDS, sources and vendors seem to come up with a variety of definitions:

- Wireless Intrusion Prevention System (WIPS): Monitors Wi-Fi networks [11]
- Network Behavior Analysis (NBA): Detect what might be associated with distributed denial of service (DDoS) attacks [11]

- Network Node Intrusion Detection System (NNIDS): Watches over each node connected to your network [12].
- Protocol-Based Intrusion Detection System (PIDS): Analyzes the HTTP or HTTPS protocol stream between your devices and the server[12].
- Etc. Put in your own use case with associated acronym here..

Some examples of IDS/IPS systems are shown in Table 1:

Features/IDS	**Real-time Monitoring**	**Log Management**	**Signature-based Detection**	**Anomaly-based Detection**	**Open Source**	**Cloud Integration**	**Free Version Available**
ManageEngine EventLog Analyzer	Yes	Yes	Yes	Yes	No	Yes	No
ManageEngine Log360	Yes	Yes	Yes	Yes	No	Yes	No
ESET Protect	Yes	Yes	Yes	Yes	No	Yes	No
Snort	Yes	Yes	Yes	Yes	Yes	No	Yes
SolarWinds Security Event Manager	Yes	Yes	Yes	Yes	No	Yes	No
OSSEC	Yes	Yes	Yes	Yes	Yes	No	Yes
Gatewatcher AIonIQ	Yes	Yes	Yes	Yes	No	Yes	No
CrowdSec	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Suricata	Yes	Yes	Yes	Yes	Yes	No	Yes
Zeek	Yes	Yes	Yes	Yes	Yes	No	Yes
Security Onion	Yes	Yes	Yes	Yes	Yes	No	Yes
AIDE	Yes	No	No	Yes	Yes	No	Yes

Table 1: *Comparison of different IDS/IPS products, data from [13]*

4 Conclusion

Web Application Firewalls (WAF) and Intrusion Detection Systems (IDS) are important sources of data to the security operation. IDS's and WAF's can help mitigate some of the risk associated with moving infrastructure and company functions to cloud based web applications. With applications being deployed on 'someone else's computer' often using 'someone else's code', it is however important to monitor the traffic and protect the applications. By limiting data infrastructure to use HTTP/S traffic, the attack surface actually gets smaller and tools can be simplified and made more efficient (WAF is an example hereof). Most services could be made web based, managers and most enterprise support functions do not need computers with advanced OS systems, but could do all work on devices with a web browser only. The security can then be shifted more towards Identity and Access Management (IAM) and a focus on social engineering attacks.

References

[1] *What is DevSecOps?* `https://www.sonatype.com/resources/articles/what-is-devsecops`. Accessed: 2024-9-10.

[2] *OWASP Top Ten*. `https://owasp.org/www-project-top-ten/`. Accessed: 2024-9-10.

[3] *Top 10 Best WAF Solutions*. `https://www.openappsec.io/post/best-waf-solutions`. Accessed: 2024-9-12.

[4] *Web Application Firewall documentation*. `https://learn.microsoft.com/en-us/azure/web-application-firewall/`. Accessed: 2024-9-12.

[5] *AWS WAF*. `https://aws.amazon.com/waf/`. Accessed: 2024-9-12.

[6] *Cloudflare WAF*. `https://www.cloudflare.com/application-services/products/waf/`. Accessed: 2024-9-12.

[7] *AppTrana - WAF*. `https://www.indusface.com/web-application-firewall.php`. Accessed: 2024-9-12.

[8] *open-appsec*. `https://www.openappsec.io/`. Accessed: 2024-9-12.

[9] *OWASP Coraza Web Application Firewall*. `https://owasp.org/www-project-coraza-web-application-firewall/`. Accessed: 2024-9-12.

[10] *SNORT.org*. `https://www.snort.org/`. Accessed: 2024-9-13.

[11] *Fortinet, IPD Definitions*. `https://www.fortinet.com/resources/cyberglossary/what-is-an-ips`. Accessed: 2024-9-12.

[12] *Helixstorm: Understanding the 5 Types of Intrusion Detection Systems*. `https://www.helixstorm.com/blog/types-of-intrusion-detection-systems/`. Accessed: 2024-9-12.

[13] *Comparitech: Intrusion Detection Systems Explained: 12 Best IDS Software Tools Reviewed*. `https://www.comparitech.com/net-admin/network-intrusion-detection-tools/`. Accessed: 2024-9-12.

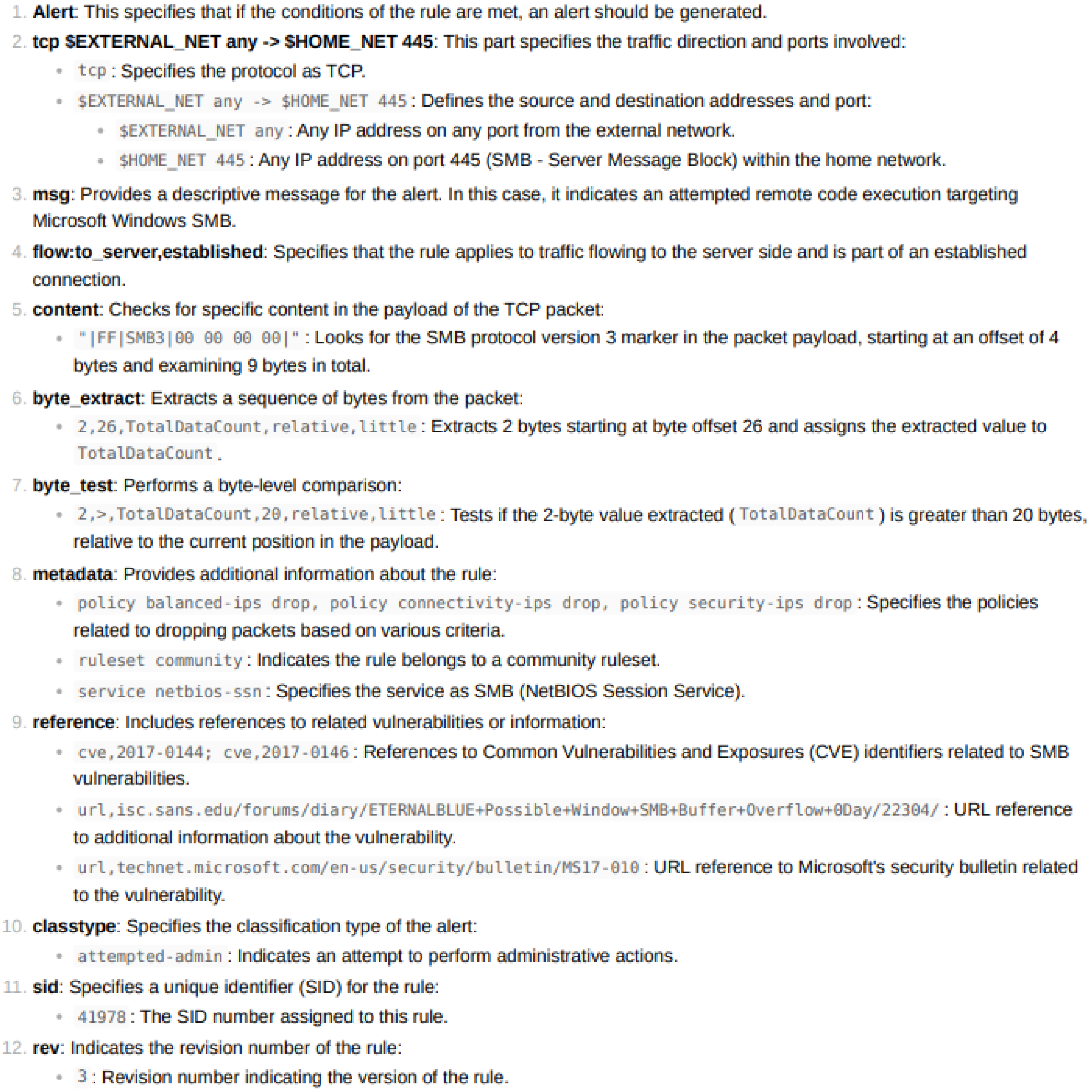
[14] *Snort SID 1:41978*. `https://www.snort.org/rule_docs/1-41978`. Accessed: 2024-9-12.

Appendices

A SNORT rule for "EternalBlue", SID 41978 see[14]

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (msg:"OS-WINDOWS
    Microsoft Windows SMB remote code execution attempt"; flow:
    to_server,established; content:"|FF|SMB3|00 00 00 00|"; depth:9;
    offset:4; byte_extract:2,26,TotalDataCount,relative,little;
    byte_test:2,>,TotalDataCount,20,relative,little; metadata:policy
    balanced-ips drop, policy connectivity-ips drop, policy security-
    ips drop, ruleset community, service netbios-ssn; reference:cve
    ,2017-0144; reference:cve,2017-0146; reference:url,isc.sans.edu/
    forums/diary/ETERNALBLUE+Possible+Window+SMB+Buffer+Overflow+0Day
    /22304/; reference:url,technet.microsoft.com/en-us/security/
    bulletin/MS17-010; classtype:attempted-admin; sid:41978; rev:3;)
```

I had ChatGPT write out following explanation to the rule, see Figure 2



1. **Alert**: This specifies that if the conditions of the rule are met, an alert should be generated.
2. **tcp $EXTERNAL_NET any -> $HOME_NET 445**: This part specifies the traffic direction and ports involved:
   - `tcp` : Specifies the protocol as TCP.
   - `$EXTERNAL_NET any -> $HOME_NET 445` : Defines the source and destination addresses and port:
     - `$EXTERNAL_NET any` : Any IP address on any port from the external network.
     - `$HOME_NET 445` : Any IP address on port 445 (SMB - Server Message Block) within the home network.
3. **msg**: Provides a descriptive message for the alert. In this case, it indicates an attempted remote code execution targeting Microsoft Windows SMB.
4. **flow:to_server,established**: Specifies that the rule applies to traffic flowing to the server side and is part of an established connection.
5. **content**: Checks for specific content in the payload of the TCP packet:
   - `"|FF|SMB3|00 00 00 00|"` : Looks for the SMB protocol version 3 marker in the packet payload, starting at an offset of 4 bytes and examining 9 bytes in total.
6. **byte_extract**: Extracts a sequence of bytes from the packet:
   - `2,26,TotalDataCount,relative,little` : Extracts 2 bytes starting at byte offset 26 and assigns the extracted value to `TotalDataCount` .
7. **byte_test**: Performs a byte-level comparison:
   - `2,>,TotalDataCount,20,relative,little` : Tests if the 2-byte value extracted (`TotalDataCount`) is greater than 20 bytes, relative to the current position in the payload.
8. **metadata**: Provides additional information about the rule:
   - `policy balanced-ips drop, policy connectivity-ips drop, policy security-ips drop` : Specifies the policies related to dropping packets based on various criteria.
   - `ruleset community` : Indicates the rule belongs to a community ruleset.
   - `service netbios-ssn` : Specifies the service as SMB (NetBIOS Session Service).
9. **reference**: Includes references to related vulnerabilities or information:
   - `cve,2017-0144; cve,2017-0146` : References to Common Vulnerabilities and Exposures (CVE) identifiers related to SMB vulnerabilities.
   - `url,isc.sans.edu/forums/diary/ETERNALBLUE+Possible+Window+SMB+Buffer+Overflow+0Day/22304/` : URL reference to additional information about the vulnerability.
   - `url,technet.microsoft.com/en-us/security/bulletin/MS17-010` : URL reference to Microsoft's security bulletin related to the vulnerability.
10. **classtype**: Specifies the classification type of the alert:
    - `attempted-admin` : Indicates an attempt to perform administrative actions.
11. **sid**: Specifies a unique identifier (SID) for the rule:
    - `41978` : The SID number assigned to this rule.
12. **rev**: Indicates the revision number of the rule:
    - `3` : Revision number indicating the version of the rule.

Figure 2: *Explanation for the SID 41978 snort Rule, generated using ChatGPT*