CS161 Fall 2026 — Introduction to Computer Security — Discussion 7

Q1 *DNS Walkthrough* **(3 points)**

Your computer sends a DNS request for www.google.com'‘

Q1.1 (1 point) Assume the DNS resolver receives back the following reply:

```
com. NS a.gtld-servers.net
a.gtld-servers.net A 192.5.6.30
```

Describe what this reply means and where the DNS resolver would look next.

Solution: The IP address for "www.google.com" is not known. However, "a.gtld-servers.net" is a name server for .com, and that is where the resolver should ask next, at the IP address 192.5.6.30.

Q1.2 (1 point) If an off-path adversary wants to poison the DNS cache, what values does the adversary need to guess?

Solution: The adversary will need to guess the identification number (16 bits). Some resolvers even randomize source ports.

The reason an off-path attack is difficult is because the ID (and port numbers) have to match exactly, but once the legitimate reply reaches the resolver and is cached, the server is no longer vulnerable to the poisoning attempts.

Q1.3 (1 point) Why do we not use TCP for DNS? Why not use TLS to make the DNS connection secure?

Solution: DNS uses UDP instead of TCP because UDP does not require a 3-way handshake and this leads to better performance.

DNS is designed to be lightweight and TLS adds a lot of overhead with encrypting and decrypting packets.

Furthermore, we do not know which name servers to trust and TLS provides no protection against that. This is a fundamental difference between object security and channel security.

There is a DNS-over-HTTPS protocol that can be used today and is becoming increasingly popular.

This content is protected and may not be shared, uploaded, or distributed.

Q2 *DNS* **(1 points)**

Q2.1 (1 point) An attacker wants to poison the local DNS resolver's cache using the Kaminsky attack. We assume that the resolver does not use source port randomization, so the attacker will likely succeed.

Using the Kaminsky attack, the attacker wants to poison the resolver's cache for the domain `www.berkeley.edu`.

The attacker constructs a website with lots of images as follows:

```
<img src="http://fake1.berkeley.edu">
<img src="http://fake2.berkeley.edu">
<img src="http://fake3.berkeley.edu">
<img src="http://fake4.berkeley.edu">
                ...
```

◇ **Question:** If the attacker wants the resolver to accept their spoofed response, the transaction id (16-bit value) must be the same as the resolver's request transaction id. If the attacker can only send k responses before the name sever responds, what is the probability that at least one of the attacker's responses will be accepted by the resolver?

Solution: Since the transaction ID is a 16-bit value, there are

$$2^{16}$$

possible IDs. The probability that one of the k attacker responses has the correct ID is

$$\frac{k}{2^{16}}$$

◇ **Question:** If the attacker can send multiple responses which use each possible transaction id, why would the attacker want to load many, even thousands of images on their website rather than just a few.

Solution: The attacker is racing to send a response with the valid transaction id to the resolver. If the name server responds first, the attacker can not poison the cache. Therefore, the attacker embeds thousands of images on their website so that they have that many chances to get the correct response to the resolver before the name server.

◇ **Question:** Assuming that the attacker has constructed a website with the image tags given above, can the attacker poison the resolver's cache for the domain www.google.com?

Solution: No, due to Bailiwick checking. The sources of the images use the name server berkeley.edu and the resolver only accepts records if they are in the name server's zone. www.google.com is not in the name server's zone of berkeley.edu.

This content is protected and may not be shared, uploaded, or distributed.

Q3 *Intrusion Detection* **(8 points)**

FooCorp is deciding which intrusion detection *method* to employ in a few target scenarios. In the following parts, consider which of the intrusion detection methods learned in class would be most appropriate (NIDS, HIDS, or logging), and justify why.

Q3.1 (1 point) FooCorp is hosting a web application over HTTPS and needs to detect any use of blacklisted characters in real time.

Solution: FooCorp should use a HIDS. While the style of detection is ideal for a NIDS, it is useless in this specific scenario because TLS hinders a NIDS from reading the application data.

Q3.2 (1 point) FooCorp is hosting a web application over HTTP and wants to pass all user traffic through an anomaly detection algorithm (which uses some computationally expensive machine learning). The web application needs to have low latency when many users are online during the day.

Solution: FooCorp should use logging with overnight analysis. Using a NIDS or HIDS would cause the web application to have increased latency since running the analysis is expensive.

Q3.3 (1 point) FooCorp uses the Simple Mail Transfer Protocol (SMTP) for email and wants to be able to quickly detect phishing attacks against any of their internal computers. SMTP runs on port 25 and is unencrypted.

Solution: FooCorp should use a NIDS here. They only need one device to be able to monitor the whole corporation and it will be real-time detection.

Q3.4 (1 point) FooCorp doesn't trust its employees and sets-up a NIDS to monitor their traffic. However, many employees use TLS, hindering what can be monitored.

FooCorp decides to turn their NIDS into a *Man-in-the-Middle*, giving it a certificate that all the employee's computers trust. Whenever an employee visits a website they complete a TLS handshake with the NIDS, the NIDS connects to the requested website using TLS, and any traffic between the employee and website is forwarded across the two TLS links by the NIDS.

Which security principle does this violate? Describe everything an attacker can do if they compromise the NIDS.

Solution: Separation of privilege. The NIDS has complete control over the contents, integrity, and authenticity of any connection.

The only certificate being verified by the client is the NIDS's so an attacker can reroute hosts to a malicious website and they wouldn't be able to tell. In other words, all the guarantees of TLS are lost.

The NIDS essentially has the same power as a MiTM with HTTP. Except it's even worse since there's a false sense of security due to the presence of TLS.

This content is protected and may not be shared, uploaded, or distributed.

FooCorp now needs to decide which intrusion detection *technique* to employ in a few target scenarios. In the following parts, consider which technique would be most appropriate (signature-based, anomaly-based, specification-based, or behavioral), and justify why.

Q3.5 (1 point) FooCorp wants to detect script kiddies (hackers who primarily use publicly available tools or exploits).

Solution: Signature-based detection would be best here since you can compile a list of all the things the hacker might do.

Q3.6 (1 point) FooCorp wants to detect a seasoned hacker who crafts custom exploits for each attack.

Solution: Behavioral/anomaly-based detection will be best here since the hacker will most likely use clever, possibly novel, techniques to pull of the attack. They can most likely avoid writing code that violates specifications or matches known signatures.

Q3.7 (1 point) FooCorp wants to detect publicly-available malware that a hacker manually tweaks to avoid signature checks.

Solution: Behavioral detection will be best here since you have a good idea of how the attack works.

Q3.8 (1 point) FooCorp wants to detect any attempts by their employees to access the protected `/etc/passwd` file.

Solution: Specification detection is best here. Simply set up rules that flag any syscalls which access `/etc/passwd`.

This content is protected and may not be shared, uploaded, or distributed.