Assignment

> Student ID: 3190104783
> Name: Ou Yixin
> Date: 2022-04-05

1 DDoS

1.1 What is the difference between DoS attacks and DDoS attacks?

DoS is Denial of Service which means to control an attacking computer/device and flood victim with superfluous requests. Then the device overload victim and prevent it from fulfilling some legitimate requests.

Based on DoS, DDoS control many different attacking sources which makes it harder to stop the attack simply by blocking a single source.

1.2 How does the TCP SYN Flood attack work?

The attack principle is that the server commits resources (memory) before confirming identify of client (when client responds). The malicious device use IP Spoofing to forge SYN packets with random source IP addresses. The IP addresses won't send ACK packets and the server will therefore hold the information for each IP addresses. Then it can gradually fill up backlog queue so that the server cannot handle more connections.

1.3 How does the solution of SYN Cookies against TCP SYN Flood attacks work?

The goal of the solution of SYN Cookies is to avoid state storage on server until 3-way handshake completes. The server will store no information but send necessary states to client along with SYN-ACK and the client has to send these states back to server along with ACK. If a malicious device wants a TCP SYN Flood, the fake IP address will not send the information back to the server so the server will not store the information of the client.

1.4 How does the DNS Amplification Attack work? How to defend against it?

A DNS amplification can be broken down into four steps:

- The attacker uses a compromised endpoint to send UDP packets with spoofed IP addresses to a DNS recursor. The spoofed address on the packets points to the real IP address of the victim.
- Each one of the UDP packets makes a request to a DNS resolver, often passing an argument such as “ANY” in order to receive the largest response possible.
- After receiving the requests, the DNS resolver, which is trying to be helpful by responding, sends a large response to the spoofed IP address.
- The IP address of the target receives the response and the surrounding network infrastructure becomes overwhelmed with the deluge of traffic, resulting in a denial-of-service.

There are two ways to defend against DNS amplification attack:

- Reduce the total number of open DNS resolvers
- Source IP verification – stop spoofed packets leaving network

2 DDoS

2.1 How does Memcached attack work?

A memcached attack occurs in 4 steps:

- An attacker implants a large payload of data on an exposed memcached server.
- Next the attacker spoofs an HTTP GET request with the IP address of the targeted victim.
- The vulnerable memcached server that receives the request, which is trying to be helpful by responding, sends a large response to the target.
- The targeted server or its surrounding infrastructure is unable to process the large amount of data sent from the memcached server, resulting in overload and denial-of-service to legitimate requests.

2.2 What is the difference between HTTP Flood and Fragmented HTTP Flood?

Devices with a valid IP are used to establish a valid HTTP connection with a web server. Then, HTTP packets are split into tiny fragments and sent to the target as slowly as it allows before it times out. This method allows the attackers to keep a connection active for a long time without alerting any defense mechanisms.

2.3 Why is Fragmented HTTP Flood relatively more challenging to detect?

An attacker can use one device to initiate several undetected, extended and resource consuming sessions. Popular web servers like Apache do not have effective timeout mechanisms. This is a DDoS security loophole that can be exploited with a few devices to stop web services.

2.4 How does Ingress Filtering?

Packets enter internet via ISP and ingress filtering policy is that ISP only forwards packets with legitimate source IP.

2.5 How does IP Traceback work?

IP Traceback change routers to record info in packets. Router adds its own IP address to packetand the victim reads path from packet. Another strategy is for the router to do edge sampling with a certain probability, so that only two fields(edge and distance) need to be appended to the packet.

3 Secure Routing

3.1 What are the key features of the five typical delivery schemes?

Unicast delivers a message to a single specific node.
Broadcast delivers a message to all nodes in the network.
Multicast delivers a message to a group of nodes that have expressed interest in receiving the message.
Anycast delivers a message to any one out of a group of nodes, typically the one nearest to the source.
Geocast delivers a message to a group of nodes based on geographic location.

3.2 What is the framework of the Dijkstra algorithm?

Dijkstra algorithm works by first making the node to every other node's distance to be infinity. Then starting to the beginning node, update every node's distance. Then pick the node with the shortest distance. Find the neighbors of it and update the info. Since every time we focus on the nearest node, so the algorithm works.

3.3 What is the framework of the Bellman Ford algorithm?

The Bellman Ford algorithm works by first making the node to every other node’s distance to be infinity. Then starting to the beginning node, update every node’s distance. Unlike Dijkstra algorithm, it doesn’t pick the node with the shortest distance each time, but update every edges. So it can handle graphs with negative weights, but it costs more time.

3.4 How does prefix hijacking work?

The requests from clients are directed to certain servers specified by the prefix. The autonomous system can be hijacked and mislead the prefix. For example, when the user wants to connect to YouTube, they wants to visit 208.65.153.0. And YouTube annouced it is 208.65.153.0/22. However, the AS is hijacked and the adversary can annouce it is YouTube with 208.65.153.0/24. So the client visits the wrong server.

3.5 How does RPKI work? Why is it insufficient for secure routing?

RPKI is a certified mapping between IP prefixes and the Ases that own them. With RPKI, ISP can check the RPKI and see that someone is not a valid originator of a certain prefix and choose the correct path to the prefix.

4 Anonymous Communication

4.1 Why is current Internet communication vulnerable to anonymity or privacy leakage?

Because though the message is encrypted, the IP addresses are not. For users to communicate via internet, their devices assigned with IP addresses, which are usually fixed within a communication session or more. This can be used to infer critical privacy of users. So who is communicating?who are you talking to?what type of activities?what type of information? can all be known.

4.2 In which scenarios do users require the communication anonymity or privacy as concerned in sub question a?

Scenarios are like unmonitored access to health and medical information, or preservation of democracy: anonymous election/jury, or Censorship circumvention: anonymous access to otherwise restricted information.

4.3 How to use proxies to secure communication anonymity? What are the possible limitations?

A user can send message to the proxy and the proxy relay it to the receiver. The true routing is inside the application layer.

The possible limitations are that requires of trusted third party, proxy may release logs, or sell them,or blackmail sender and anonymity largely depends on the location of attacker.

4.4 How does Onion Routing provide a better guarantee for anonymity?

Because the overlay communication is also anonymous. It seems like many proxies are used to send the message. And each time a union router only knows the information of its neighbors. Message is encrypted serveral times.

4.5 How to infer anonymity or privacy of Onion Routing traffic?

We can go inside the Onion Routing traffic and find correlations such as path selection attack, counting attack, low latency attack and cross site attack.

5 Web Security

5.1 How does Same Origin Policy work?

Same Origin Policy prevents a malicious site from spying on or tampering with user information or interactions with other websites.

Policy 1: Each site in the browser is isolated from all others

Policy 2: Multiple pages from the same site are not isolated

5.2 How does SQL Injection work? How to defend against it?

SQL injection is a code injection technique used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution.

The fundamental cause of SQL injection is mixing data and code.

The fundamental solution of SQL Injection is to separate data and code such as using Prepared Statement.

5.3 Please refer to the slides or search online and provide two concrete examples of SQL Injection.

Example 1:

`user = " ' or 1=1 -- "`

Then script does: `ok = execute( SELECT * FROM Users  WHERE user= ' ' or 1=1 -- … )`

The “`--`” causes the rest of line to be ignored, so OK is always true and login succeeds.

Example 2:

`user = " ' ; INSERT INTO TABLE Users (‘attacker’,  ‘attacker secret’); -- "`

Then script does: `ok = execute( SELECT * FROM Users  WHERE user= ' ' ; INSERT INTO TABLE Users ('attacker',  'attacker secret'); -- … )`

Create another account with password.

6 Web Security

6.1 How does a DNS hijacking attack affect network security?

DNS hijacking attack is the DNS queries are incorrectly resolved in order to unexpectedly redirect users to malicious sites. When the user enter a valid website, he or she is directed to a fake one and the attackers may phishing or do other things to steal the user’s information.

6.2 In HTTPS, how does a user verify a certificate for determining the authenticity of the website it connects to?

A user receives the domain name, public key signed by a CA. Then the user needs to decide whether to trust the CA and it’s signature or not. The user has a local built-in certificate store and use the CA’s public key and signature to test whether the CA that website claimed is indeed the true CA.

6.3 Please provide a concrete example to showcase CSRF.

Suppose an application contains a function that lets the user change the email address on their account. When a user performs this action, they make an HTTP request like the following:

```
POST /email/change HTTP/1.1
Host: vulnerable-website.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Cookie: session=yvthwsztyeQkAPzeQ5gHgTvlyxHfsAfE

email=wiener@normal-user.com
```

This meets the conditions required for CSRF:

- The action of changing the email address on a user's account is of interest to an attacker. Following this action, the attacker will typically be able to trigger a password reset and take full control of the user's account.
- The application uses a session cookie to identify which user issued the request. There are no other tokens or mechanisms in place to track user sessions.
- The attacker can easily determine the values of the request parameters that are needed to perform the action.

With these conditions in place, the attacker can construct a web page containing the following HTML:

```
<html>
    <body>
        <form action="https://vulnerable-website.com/email/change" method="POST">
            <input type="hidden" name="email" value="pwned@evil-user.net" />
        </form>
        <script>
            document.forms[0].submit();
        </script>
    </body>
</html>
```

If a victim user visits the attacker's web page, the following will happen:

- The attacker's page will trigger an HTTP request to the vulnerable web site.
- If the user is logged in to the vulnerable web site, their browser will automatically include their session cookie in the request.
- The vulnerable web site will process the request in the normal way, treat it as having been made by the victim user, and change their email address.

6.4 Please provide two concrete examples to showcase Stored XSS and Reflective XSS.

Stored cross-site scripting arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.

Suppose a website allows users to submit comments on blog posts, which are displayed to other users. Users submit comments using an HTTP request like the following:

```
POST /post/comment HTTP/1.1
Host: vulnerable-website.com
Content-Length: 100

postId=3&comment=This+post+was+extremely+helpful.&name=Carlos+Montoya&email=carlos%40n
ormal-user.net
```

After this comment has been submitted, any user who visits the blog post will receive the following within the application's response:

```
<p>This post was extremely helpful.</p>
```

Assuming the application doesn't perform any other processing of the data, an attacker can submit a malicious comment like this:

```
<script>/* Bad stuff here... */</script>
```

Within the attacker's request, this comment would be URL-encoded as:

```
comment=%3Cscript%3E%2F*%2BBad%2Bstuff%2Bhere...%2B*%2F%3C%2Fscript%3E
```

Any user who visits the blog post will now receive the following within the application's response:

```
<p><script>/* Bad stuff here... */</script></p>
```

The script supplied by the attacker will then execute in the victim user's browser, in the context of their session with the application.

Reflected cross-site scripting arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.

Suppose a website has a search function which receives the user-supplied search term in a URL parameter:

```
https://insecure-website.com/search?term=gift
```

The application echoes the supplied search term in the response to this URL:

```
<p>You searched for: gift</p>
```

Assuming the application doesn't perform any other processing of the data, an attacker can construct an attack like this:

```
https://insecure-website.com/search?term=<script>/*+Bad+stuff+here...+*/</script>
```

This URL results in the following response:

```
<p>You searched for: <script>/* Bad stuff here... */</script></p>
```

If another user of the application requests the attacker's URL, then the script supplied by the attacker will execute in the victim user's browser, in the context of their session with the application.

7 Email Security

7.1 Please describe common threats against Email security.

- Authenticity-related Threats: could result in unauthorized access to an email system
- Integrity-related Threats: could result in unauthorized modification of email content
- Confidentiality-related Threats: could result in unauthorized disclosure of sensitive information
- Availability-related Threats: could prevent end users from being able to send or receive email

7.2 How should an Email be protected to support both Authentication and Confidentiality?

The procedure is as follows:

1. The sender sign the message with RSA/SHA-256 using his/her private key.
2. Generate a one-time secret key to encrypt the message.

3. Encrypt the one-time secret key with RSA using the receiver’s public key and append the result to the massage.
4. The receiver decrypt the one-time secret key with RSA using his/ger private key.
5. Decrypt the message with the one-time secret key.
6. Verify the signature with RSA/SHA-256 using the sender’s public key.

7.3 Please describe the differences among DANE, SPF, and DKIM.

DANE is DNS-based Authentication of Name Entities. DANE allow X.509 certificates to be bound to DNS names using DNSSEC.

SPF is Sender Policy Framework. ADMDs (Administrative Management Domains) publish SPF records in DNS specifying which hosts/IP-addresses are permitted to use their names. Receivers use the published SPF records to test the authorization of sending Mail Transfer Agents (MTAs) using a given “HELO” or “MAIL FROM” identity during a mail transaction.

DKIM is DomainKeys Identified Mail. DKIM sign email message by a private key of the administrative domain from which the email originates. At the receiving end, the MDA can access the corresponding public key via a DNS and verify the signature, thus authenticating that the message comes from the claimed administrative domain

8 Traffic Analysis

8.1 Please describe the properties of the four types of commonly used Firewall.

Packet Filtering Firewall applies a set of rules to each incoming and outgoing IP packet.

Stateful Inspection Firewall examines both packets and their context.

Application Proxy Firewall acts as a relay of application-level traffic.

Circuit-Level Proxy Firewall acts as a relay of TCP segments without examining the contents.

8.2 What are the differences among Firewall, IDS, and IPS?

Firewall supports active filtering.

IDS provides only passive monitoring.

IPS is an extension of IDS to attempt to block or prevent detected malicious activity.

8.3 Please list commonly used methods for obfuscating traffic to evade detection.

- Encrypt traffic to hide payloads
- Use proxy to hide entire packets
- Introduce noise traffic to hide patterns

9 Open Question: Authentication Efficiency

Consider a time consuming authentication scenario where a database records all secret keys of a large number of users. When the system authenticates a user, it first issues a challenge message to the user. The user then uses his/her key to encrypt the challenge and then returns the encrypted challenge to the system. The system then encrypts the challenge using one key in the database after another and compares the result with the received encrypted message. Once a match is found, the system accepts the user. Otherwise, the user is denied. This authentication protocol surely takes a lot of time and computation. Design a possible solution to speed up the authentication process.

Besides all secret keys of users, the database can also record the fixed challenge message encrypted by each secret key in a hash table. When the system authenticates a user, it first issues the fixed challenge message to the user. The user then uses his/her key to encrypt the challenge and then returns the encrypted challenge to the system. The system then query the encrypted message in hash table. Once the hash hits, the system accepts the user. Otherwise, the user is denied.

10 Share your thoughts on the course

10.1 To what extent do you devote your time and energy to labs? How do you overcome the associated challenges?

Each lab takes me an average of four days to complete.

When I encounter the associated challenges, I will first turn to searching online for help, and if I still can't solve it, I will discuss the problem with my partners.

10.2 Do you think that you have gradually cultivated a research/security mindset? What is the most useful idea that you learned during this process?

Yes, I think I have gradually cultivated a research/security mindset. The most useful idea is how to solve a problem in a field I haven't studied much before.

10.3 Provide an example to showcase how you leverage that useful idea to facilitate problem solving in study or life.

When doing research in a field I haven't studied much before, the best way is to read several reviews of the literature. Then I will follow the clues in the paper's citation list to focus on the more highly cited work in the field.

11 Design a question that you think is feasible as an exam question

11.1 Which topic among the lectures you would like to consider?

I would like to choose the topic of DDoS.

11.2 Describe a (sufficiently complex) question.

What’s the difference between symmetric DDoS attack and asymatric DDoS attack? Please provide one common attack as example of each of them.

11.3 Provide also a correct sample solution, thanks.

In symmetric DDoS attack, the amount of bandwidth the targeted device consumes is simply the sum of the total traffic sent from each attacker/bot, such as TCP SYN Flood.

In asymatric DDoS attack, a relatively small number or low levels of resources are required by an attacker to cause a significantly greater number or higher level of target resources to malfunction or fail, such as Smurf Attack.