ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Network Security Group, Department of Computer Science, ETH Zurich

Information Security Lab: Module 05

Exercise Sheet week 2

TAs: Christelle Gloor, Felix Stöger

DDoS Attacks & Defenses

Fall 2022

Emails: christelle.gloor@inf.ethz.ch, felix.stoeger@inf.ethz.ch

Denial-of-service attacks are unique in their near-universal applicability. Every system that exposes some service to the Internet is potentially vulnerable to some type of this attack. To defend effectively against DDoS, all the protocols, applications, implementations, hardware, and the network itself must be DDoS-resilient.

This exercise sheet aims to provide some insights into how different layers of the networking stack can defend against DDoS attacks.

Exercise 1: Protocol Layer

We consider a key exchange protocol that performs a simple Diffie-Hellman (DH) exchange between two parties A and B with authentication of the responder B. This type of protocol is used widely in networking, for example in virtual private network (VPN) applications such as IPsec (with its IKEv2 [2] protocol) and WireGuard [1] (which uses the Noise [3] framework for cryptographic protocols).

For the purpose of this exercise, we are stripping out many of the details of these protocols and focusing only on some parts that are relevant for the topic of DoS attacks.

The protocol can be summarized in Alice-and-Bob notation as follows:

$$A \longrightarrow B: \quad g^x \qquad (m_1)$$
$$A \longleftarrow B: \quad g^y, \{g^y\}_{\mathrm{sk_B}} \qquad (m_2)$$

where g is the generator of a DH group, and $\mathrm{sk_B}$ is the private key of B. After this message exchange, both A and B compute the session key g^{xy}.

(a) In this notation, many critical aspects of the protocol are left out, such as the computational steps necessary for the two parties. While usually implied, precise knowledge of these steps is necessary to analyze DoS resilience.

Let us assume that a VPN endpoint B has 8 CPU cores that can process requests in parallel. We assign some rough processing times to the following types of operations at B:

- Generating a DH value x or y: 2 ms
- DH exponentiation: 13 ms
- Digital signature computation: 10 ms

You may assume that all other steps take no time to process.

Finally, we assume that a request packet (m_1) uses UDP over IP as its transport (which is usually the case for VPN protocols) and therefore is about 100 bytes large.

Consider an attacker that controls a single machine at a fixed IP, but who can spoof any source addresses to send to B. **How much bandwidth is required for this attacker to completely exhaust B's computational resources?**

(b) To mitigate this vulnerability, we modify the protocol with a cookie mechanism (similar to TCP cookies):

$$
\begin{aligned}
A \longrightarrow B: &\quad g^x && (m_1)\\
A \longleftarrow B: &\quad \text{cookie} := \text{h}(g^x, \text{salt}_B) && (m_c)\\
A \longrightarrow B: &\quad g^x, \text{cookie} && (m_1')\\
A \longleftarrow B: &\quad g^y, \{g^y\}_{\text{sk}_B} && (m_2)
\end{aligned}
$$

where h is a cryptographic hash function, and salt_B refers to a secret value only known to B.

What is the problem with this approach? **Explain how the single-machine attacker can bring down the server again, despite the cookie mechanism.**

The cookie construction in (m_c) can be fixed by using the following input:

$$\text{cookie} := \text{h}(A, g^x, \text{salt}_B) \tag{1}$$

where A refers to the address of A, which corresponds to the tuple (IP, port) in the IP Internet.

This is essentially the mechanism that is used in real-world VPN protocols. Until now, we have considered a very simple adversary model where spoofing was possible arbitrarily, but as we have seen in the lecture, there are techniques to mitigate spoofing. We will explore this in the next task.

Exercise 2: Network Layer

The cookie mechanism introduced in the exercise above provides a property that is often referred to as *weak source authentication*, which essentially states "the communication partner is reachable at the address it claims to use". This is in contrast to stronger methods such as TLS client certificates or pre-shared keys, where cryptographic information is bound to an identity using out-of-band mechanisms (such as a public-key infrastructures).

(a) In environments such as mobile networks or public networks, IP addresses are not statically bound to a user. **What consequence does this have for the reliability of weak source authentication mechanisms?**

(b) Consider the IP Internet topology in Figure 1, where the VPN endpoint runs a cookie-based key exchange protocol such as the one we discussed above.

The attacker has the possibility to eavesdrop on traffic in AS 3 between the routers. In order to remain stealthy, the wiretap is passive: traffic cannot be injected, only observed. **How can this position be abused to circumvent the protections offered by the cookie mechanism of the VPN protocol?**

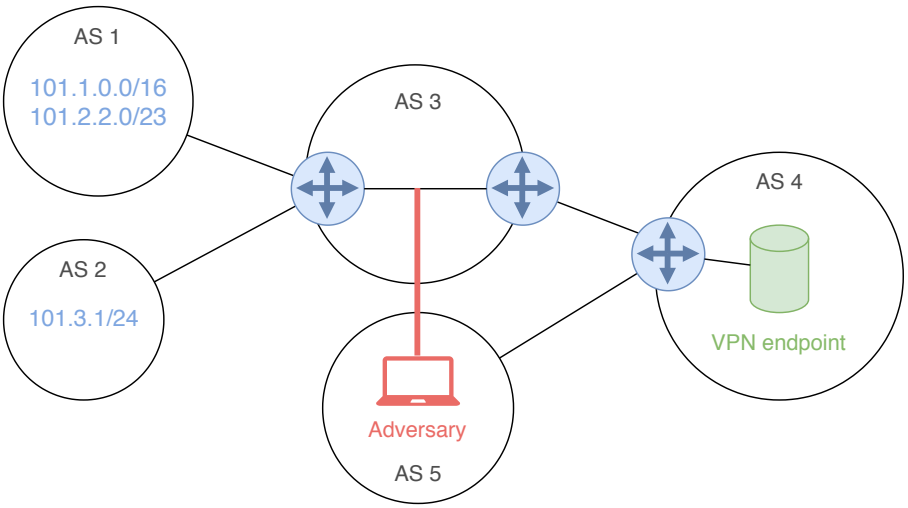

Figure 1: A network topology where the adversary has a wiretap set up. The IP ranges refer to the prefixes that the respective ASes announce.

(c) What options does the **VPN endpoint** have to defend against this attack (without changing any of the network setup)?

(d) How could the **border router** in AS 4 assist in mitigating this attack? *Hint: The attack still relies on source address spoofing*.

References

[1] Jason A. Donenfeld. WireGuard: next generation kernel network tunnel. Technical report, WireGuard, 2017.

[2] Pasi Eronen, Yoav Nir, Paul E. Hoffman, and Charlie Kaufman. Internet key exchange protocol version 2 (IKEv2). RFC 5996, September 2010.

[3] Trevor Perrin. The Noise protocol framework. `https://noiseprotocol.org/noise.pdf`, 2016.