CALIFORNIA STATE UNIVERSITY
LONG BEACH
College of Engineering

CECS 303: Networks and Network Security

Common Network Attacks (cont'd)

Chris Samayoa

Week 13 – 2nd Lecture
4/14/2022

Course Information

CALIFORNIA STATE UNIVERSITY
LONG BEACH
College of Engineering

- CECS 303
  - Networks and Network Security – 3.0 units
- Class meeting schedule
  - TuTH 5:00PM to 7:15PM
  - Lecture Room: VEC 402
  - Lab Room: ECS 413
- Class communication
  - chris.samayoa@csulb.edu
  - Cell: 562-706-2196
- Office hours
  - Thursdays 4pm-5pm (VEC-404)
  - Other times by appointment only

Objectives

- Sandworm
- Common Network Attacks – Overview
- SQL Injection Attacks
- Cross-Site Scripting (XSS)

Sandworm Hacking Group

- Previously attributed to Russian GRU's (Chief Intelligence Office) Main Centre for Special Technologies GTsST (Main Center for Special Technologies)
  - CISA, FBI, and NSA have all made this link
- Previous Malicious Cyber Activity
  - BlackEnergy – disruption of Ukrainian electricity in 2015
  - Industroyer (aka Crashoverride) – first malware designed specifically to effect electrical grids
    - Used for second outage of Ukrainian electricity in 2016
    - Main components: Malware creates backdoor for C&C servers, secondary backdoor established, launcher, four payloads specific to ICS protocols, and data wiper used to make system unbootable
  - NotPetya (2017)
- VPNFilter malware (started in 2016)
  - Attacked edge network devices
  - Used for intelligence collection and cyber attack operations (botnet)
  - Persisted through reboots
  - Included self-destruct command

Cyclops Blink

- Large-scale malware framework affecting network devices
  - Currently targeting WatchGuard devices
    - Uses previously patched authentication bypass vulnerability (CVE-2022-23176)
    - Vulnerability patched by WatchGuard in May 2021, but not specifically disclosed to public as a known vulnerability
      - Security through obscurity not being criticized
  - Essentially an updated VPNFilter malware
  - Deployed since at least June 2019
  - Persistent on reboot of device
- FBI Action
  - Recovered firmware image from compromised devices
    - Monitored traffic from infected device
  - Identified a C&C server
    - Gained access to the server and found a digital certificate that could be used to identify other C&C servers (22 in U.S. that were then seized)
  - Cleaned compromised devices found on the internet
    - Added firewall rules to then block remote access to management interface
    - Firewall rules were not made persistent

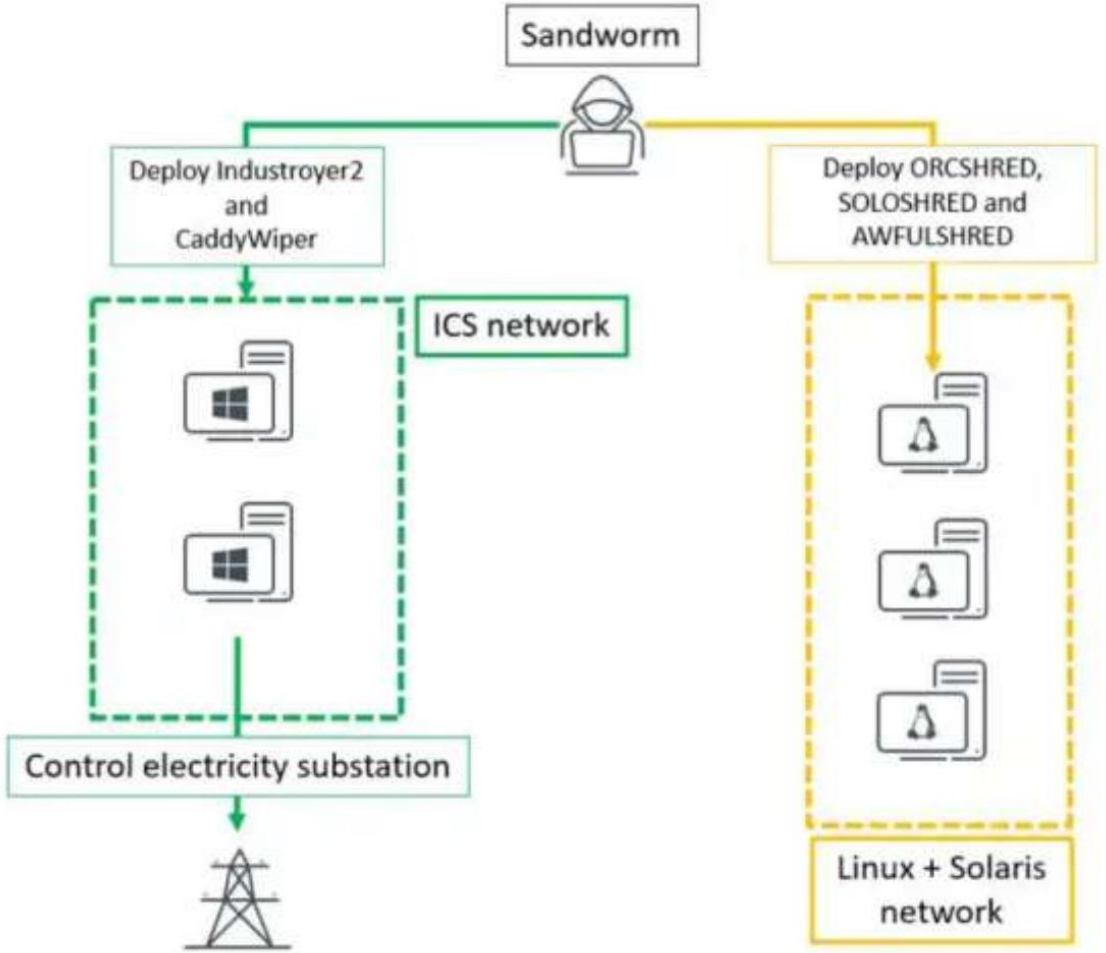
Cyclops Blink (cont'd)

- Is this an example of Grey Hat Hacking?
- References
  - https://www.csoonline.com/article/3656913/fbi-active-defense-measure-removes-malware-from-privately-owned-firewalls.html
  - https://www.ncsc.gov.uk/news/joint-advisory-shows-new-sandworm-malware-cyclops-blink-replaces-vpnfilter
  - Indicators of Compromise: https://www.ncsc.gov.uk/files/Cyclops-Blink-Malware-Analysis-Report.pdf

Industroyer 2

- General information
  - Targeted Ukrainian high-voltage electrical substations
  - Unsuccessful as it was identified before scheduled execution of malicious activity
  - ICS-capable malware and disk wipers
- Affected / targeted devices
  - Windows, Linux, and Solaris operating systems
- Goals
  - Elevate privileges
  - Move laterally within operational technology (OT) networks
  - Disrupt critical devices
  - Execute commands

Industroyer 2 (cont'd)



Source: https://www.welivesecurity.com/2022/04/12/industroyer2-industroyer-reloaded/

Industroyer 2 (cont'd)

Original CaddyWiper Sample

```
strcpy(s_netapi32, "netapi32.dll");
(LoadLibraryA)(s_netapi32);
Buffer = 0;
result = DsRoleGetPrimaryDomainInformation(0, DsRolePrimaryDomainInfoBasic, &Buffer);
if ( *Buffer != DsRole_RolePrimaryDomainController )
{
  (LoadLibraryA)(s_advapi32);
  strcpy(dir, "C:\\Users");
  Wipe(dir);
  strcpy(drive, "D:\\");
  for ( i = 0; i < 24; ++i )
  {
    Wipe(drive);
    ++drive[0];
  }
```



```
strcpy(s_netapi32, "netapi32.dll");
v5 = (LoadLibraryA)(s_netapi32);
strcpy(s_DsRoleGetPrimaryDomainInformation, "DsRoleGetPrimaryDomainInformation");
DsRoleGetPrimaryDomainInformation = GetProcAddress(v5, s_DsRoleGetPrimaryDomainInformation);
Buffer = 0;
DsRoleGetPrimaryDomainInformation(0, DsRolePrimaryDomainInfoBasic, &Buffer);
result = Buffer;
if ( *Buffer != DsRole_RolePrimaryDomainController )
{
  strcpy(dir, "C:\\Users");
  Wipe(GetProcAddress, dir);
  drive = '\\:D';
  max = 24;
  do
  {
    Wipe(GetProcAddress, &drive);
    LOBYTE(drive) = drive + 1;
    --max;
  }
  while ( max );
```


Sample found at Ukrainian energy provider

Source: https://www.welivesecurity.com/2022/04/12/industroyer2-industroyer-reloaded/

Industroyer 2 (cont'd)

SHA-1	Filename	ESET detection name	Description
FD9C17C35A68FC505235E20C6E50C622AED8DEA0	108_100.exe	Win32/Industroyer.B	Industroyer2
6FA04992C0624C7AA3CA80DA6A30E6DE91226A16	zrada.exe	Win32/Agent.AECG	ArguePatch
9CE1491CE69809F92AE1FE8D4C0783BD1D11FBE7	pa.pay	N/A	TailJump (Encrypted CaddyWiper)
0090CB4DE31D2D3BCA55FD4A36859921B5FC5DAE	link.ps1	PowerShell/HackTool.Agent.AH	Script which enumerates GPO
D27D0B9BB57B2BAB881E0EFB97C740B7E81405DF	sc.sh	Linux/Agent.PC trojan	OrcShred (Linux worm)
3CDBC19BC4F12D8D00B81380F7A2504D08074C15	wobf.sh	Linux/KillFiles.C trojan	AwfulShred (Linux wiper)
8FC7646FA14667D07E3110FE754F61A78CFDE6BC	wsol.sh	Linux/KillFiles.B trojan	SoloShred (Solaris wiper)

Source: https://www.welivesecurity.com/2022/04/12/industroyer2-industroyer-reloaded/

Pipedream / Incontroller

- General information
  - Another ICS Malware
  - Again a modular ICS attack framework
  - Dragos coined it Pipedream | Mandiant coined it Incontroller
  - Attacker must already have access to OT network
  - Attributed to Chernovite (Russian-linked) – not confirmed
- Affected / targeted devices
  - Schneider Electric programmable logic controllers (PLCs)
  - OMRON Sysmac NEX PLCs
  - Open Platform Communications Unified Architecture (OPC UA) server
  - Windows-based workstations / servers
    - If using ASRock motherboard driver with known vulnerabilities
- https://www.cisa.gov/uscert/ncas/alerts/aa22-103a

Objectives

- Sandworm
- Common Network Attacks – Overview
- SQL Injection Attacks
- Cross-Site Scripting (XSS)

Common Network Attacks

- Malware
  - Malicious software that is used to exploit devices at the expense of victim resources
- Distributed Denial of Service Attack (DDOS)
  - An attack where multiple (typically compromised) systems attack a target with the goal being to overwhelm the resource and make it unavailable for use
- SQL Injection Attacks
  - Attackers can construct a web request that provides unintended access to database resources
  - Can be used to create, modify, delete, or extract data from a database
- Cross-site Scripting (XSS) Attack
  - Attacker injects a malicious script into a trusted website
  - Injected script will then be delivered to a victim's web browser
  - Used to spread malware, steal credentials, or steal user sessions
- Man-in-the-Middle Attack
  - Attacker intercepts communications between two or more parties to intercept data
- DNS Tunneling
  - Command-and-control tactic that uses DNS queries to go undetected
- Email Attacks

Objectives

- Sandworm
- Common Network Attacks – Overview
- SQL Injection Attacks
- Cross-Site Scripting (XSS)

SQL Injection

- Purpose
  - Craft calls to web server with the intention of having malicious instructions sent to a SQL server in the backend
  - Works on dynamic SQL statements
    - Statement is generated at run time using parameters from web form or other query
- Potential Impacts
  - Database corruption
  - Authentication bypass
  - Data tampering / modification (integrity issue)
  - Data theft / exfiltration (confidentiality issue)
  - Deletion of data
  - Arbitrary code execution
  - Complete compromise of system (root access)

SQL Injection - Types

- Union-based SQL injection
  - Most popular type of SQL injection
  - Uses UNION statement to combine two ‘select’ statements to retrieve data from other tables
- Error-Based SQL Injection
  - Works on MS-SQL servers
  - Idea is to force an error that also contains the data attacker seeks
- Blind SQL Injection
  - No error messages are received from database
  - Data extracted by submitting queries to database
  - Methods:
    - Boolean-based
    - Time-based

```
SELECT columnName, columnName2 FROM tableName WHERE ID = 14 and 1=1SELECT
```

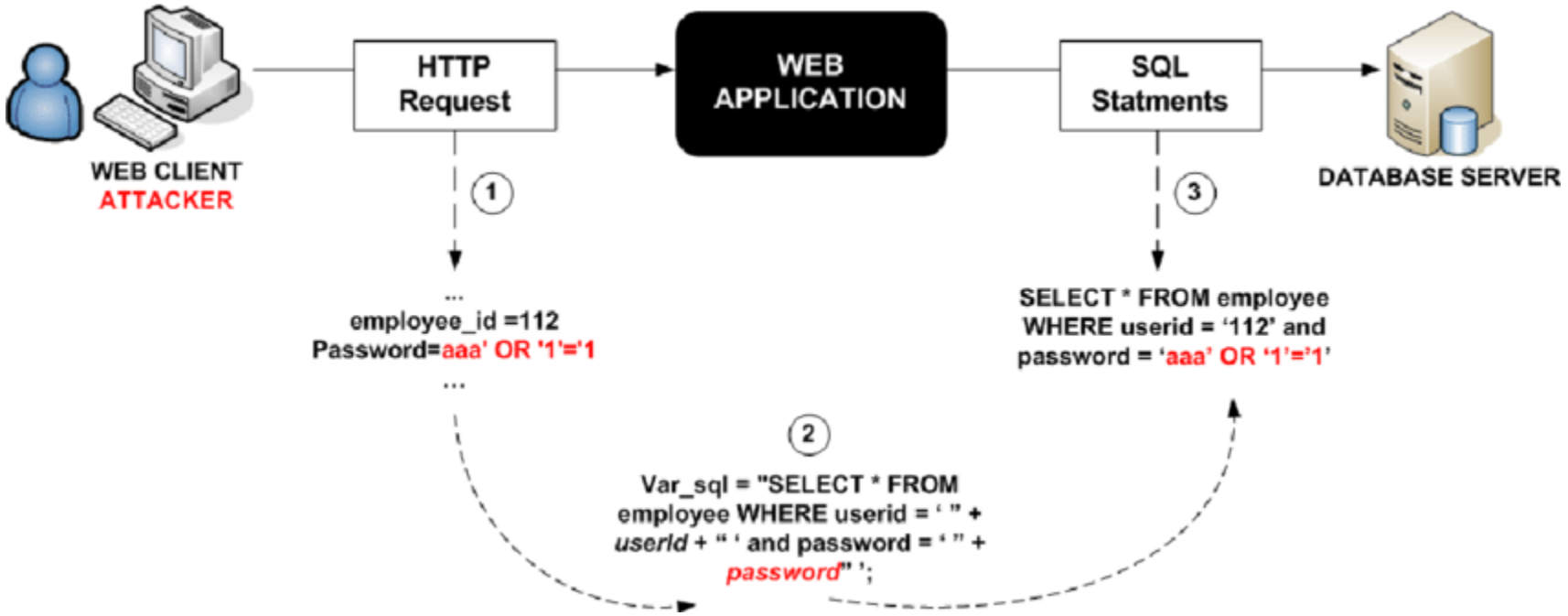
Source: https://brightsec.com/blog/sql-injection-attack/

SQL Injection - Methods

- Based on user input
  - Web application that uses a form to obtain a user's input
  - Inputs accepted without sanitizing properly
    - ➢ Leads to malicious SQL statements being injected
- Based on cookies
  - Modified cookies can "poison" database queries
  - Web application commonly load cookies from a user's browser
- Based on HTTP headers
  - Fake headers with arbitrary SQL commands can be used to inject code into a database
- Second-order SQL injection
  - Complex
  - Can lie dormant for long periods of time
    - ➢ User-supplied stat is stored by application and later incorporated into SQL queries

Source: https://brightsec.com/blog/sql-injection-attack/

SQL Injection (cont'd)



Source: https://www.researchgate.net/figure/Example-of-a-SQL-Injection-Attack_fig1_265947554

SQL Injection - Example

Simple HTML Login Form

```
<form action='index.php' method="post">

<input type="email" name="email" required="required"/>

<input type="password" name="password"/>

<input type="checkbox" name="remember_me" value="Remember me"/>

<input type="submit" value="Submit"/>

</form>
```

Source: https://www.guru99.com/learn-sql-injection-with-practical-example.html#2

SQL Injection - Example

Assume backend SQL statement is as follows:

```
SELECT * FROM users WHERE email = $_POST['email']
AND password = md5($_POST['password']);
```

Create table and user:

```
CREATE TABLE `users` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `email` VARCHAR(45) NULL,
  `password` VARCHAR(45) NULL,
  PRIMARY KEY (`id`));


insert into users (email,password) values ('m@m.com',md5('abc'));
```

SQL Injection - Example

Expected Command:

Suppose user supplies **admin@admin.sys** and **1234** as the password. The statement to be executed against the database would be:

SELECT * FROM users WHERE email = ‘admin@admin.sys’ AND password = md5(‘1234’);

Potential Exploit:

xxx@xxx.xxx’ OR 1 = 1 LIMIT 1 — ‘] would result in a true condition and the password check being commented out

SELECT * FROM users WHERE email = ‘xxx@xxx.xxx’ OR 1 = 1 LIMIT 1 — ‘] AND password = md5(‘1234’);

Source: https://www.guru99.com/learn-sql-injection-with-practical-example.html#2

SQL Injection - Example

Result from previous command:

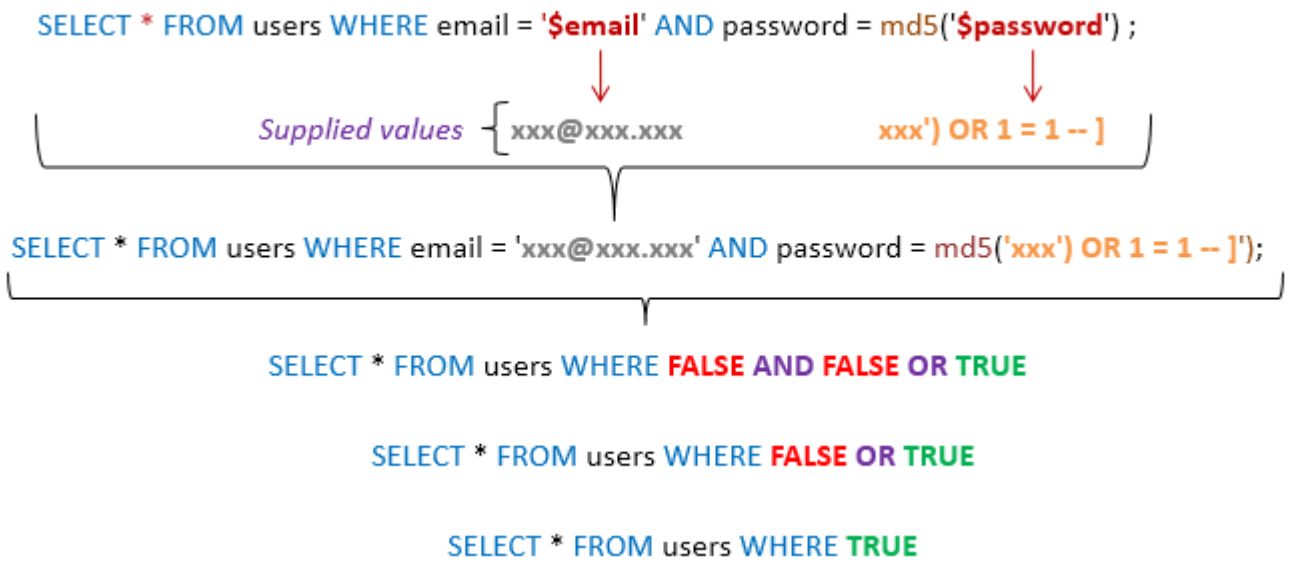
Source: https://www.guru99.com/learn-sql-injection-with-practical-example.html#2

SQL Injection – Example 2

Password value can also be used:

Password entered = xxx') OR 1 = 1 —]

SELECT * FROM users WHERE email = 'xxx@xxx.xxx' AND password = md5('xxx') OR 1 = 1 —]');



Source: https://www.guru99.com/learn-sql-injection-with-practical-example.html#2

SQL Injection - News

- Redhack group claims to have erased debt owed to government agencies (2013)
  - https://www.securityweek.com/activist-group-targets-istanbul-admin-portal-claims-have-erased-debts
  - Turkey Electricity Tranmission Company website hacked
    - Bills for Turkish citizens deleted
- 7-Eleven and others have credit card numbers stolen (2007)
  - https://www.csoonline.com/article/2124293/identity-theft-prevention-sql-injection-attacks-led-to-heartland-hannaford-breaches.html
  - 130 million credit/debit card numbers were stolen
- Fortnite user accounts (2019)
  - https://thehackernews.com/2019/01/fortnite-account-hacked.html
  - User accounts could be accessed and taken over
    - 350 million users

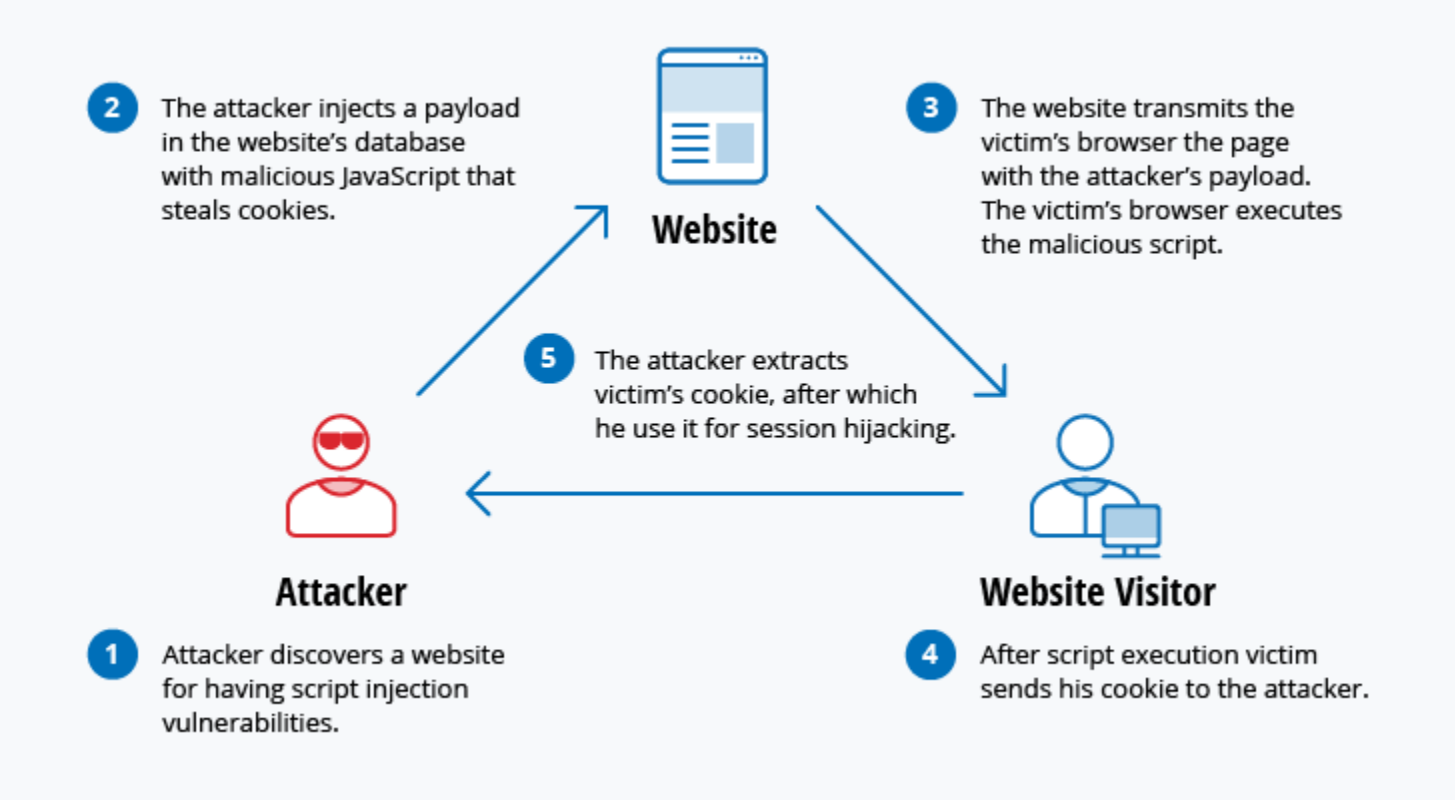
Objectives

- Sandworm
- Common Network Attacks – Overview
- SQL Injection Attacks
- Cross-site Scripting (XSS)

XSS

- Overview
  - Use third-party web resources to run scripts in a victim's web browser
  - Attacker injects malicious payload into a website's database
    - Payload delivered to victim's browser when a webpage is requested
    - JavaScript is typically used by the attacker
  - Similar to SQL injection, but target is website's users (not the web application itself)
- Prevalence
  - According to OWASP.org these are the third most common vulnerability type in 2021
    - Now counted in combination with SQL injections
- Potential Impacts
  - Steal cookies
  - Log key strokes
  - Establish remote access
  - Use machine as botnet

XSS - Example



Source: Netwrix Blog

XSS Examples

- Redhack group claims to have erased debt owed to government agencies (2013)
  - https://www.securityweek.com/activist-group-targets-istanbul-admin-portal-claims-have-erased-debts
  - Turkey Electricity Tranmission Company website hacked
    - Bills for Turkish citizens deleted
- 7-Eleven and others have credit card numbers stolen (2007)
  - https://www.csoonline.com/article/2124293/identity-theft-prevention-sql-injection-attacks-led-to-heartland-hannaford-breaches.html
  - 130 million credit/debit card numbers were stolen
- Fortnite user accounts (2019)
  - https://thehackernews.com/2019/01/fortnite-account-hacked.html
  - User accounts could be accessed and taken over
    - 350 million users