Bezpečnost

Michal Procházka
Daniel Kouřil

Proč bezpečnost?

- Zamezení úniku citlivých dat
- Zamezení neautorizovanému přístupu
- Zamezení zneužití prostředků (ftp archiv, …)

- *Zpravidla levnější než řešení následků*
  - Jinak je snadnější přijmout rizika

Zabezpečení bezpečnosti

- „Security is not a product; it‘s a process“
  - B. Schneier

- Bezpečnostní postupy a procedury se musí pravidelně revidovat
- Reflektování bezpečnostních incidentů, analýzy rizik, technologického rozvoje, technik útoků, apod.

Terminologie

- Zranitelnost vs. exploit
- Incident
- Autentizace (ne autentifikace, autentikace) vs. Autorizace
- Botnet
- Script kiddies
  - Na řadu zranitelností existují exploity (některé mohou být pastmi na samotné útočníky!)

Typický incident

- Kompromitování stroje
- (eskalace práv)
- Instalace backdoor, bota

Řešení incidentů

- Computer Security Incident Response Team (CSIRT)
- Bezpečnostní politiky
  - Stanovení zodpovědnosti
  - Stanovení procedur pro řešení incidentu a praktik pro jejich předcházení
- Bezpečnostní odborník nemusí být správce systému
- Důležitá je komunikace se světem
  - Dobrá reputace v komunitě

Obrana proti útokům

- Identifikace hrozeb a rizik
  - Silná hesla nemají smysl, pokud stroj nabízí děravé služby
- Správná konfigurace
  - Mnoho služeb je v defaultní konfiguraci špatně nastavená
- Včasná aktualizace
  - Řeší/omezí drtivou většinu útoků – pro útočníka je levnější jít o dům dál
- Monitoring
  - Centrální sběr logů
  - Sběr netflow záznamů

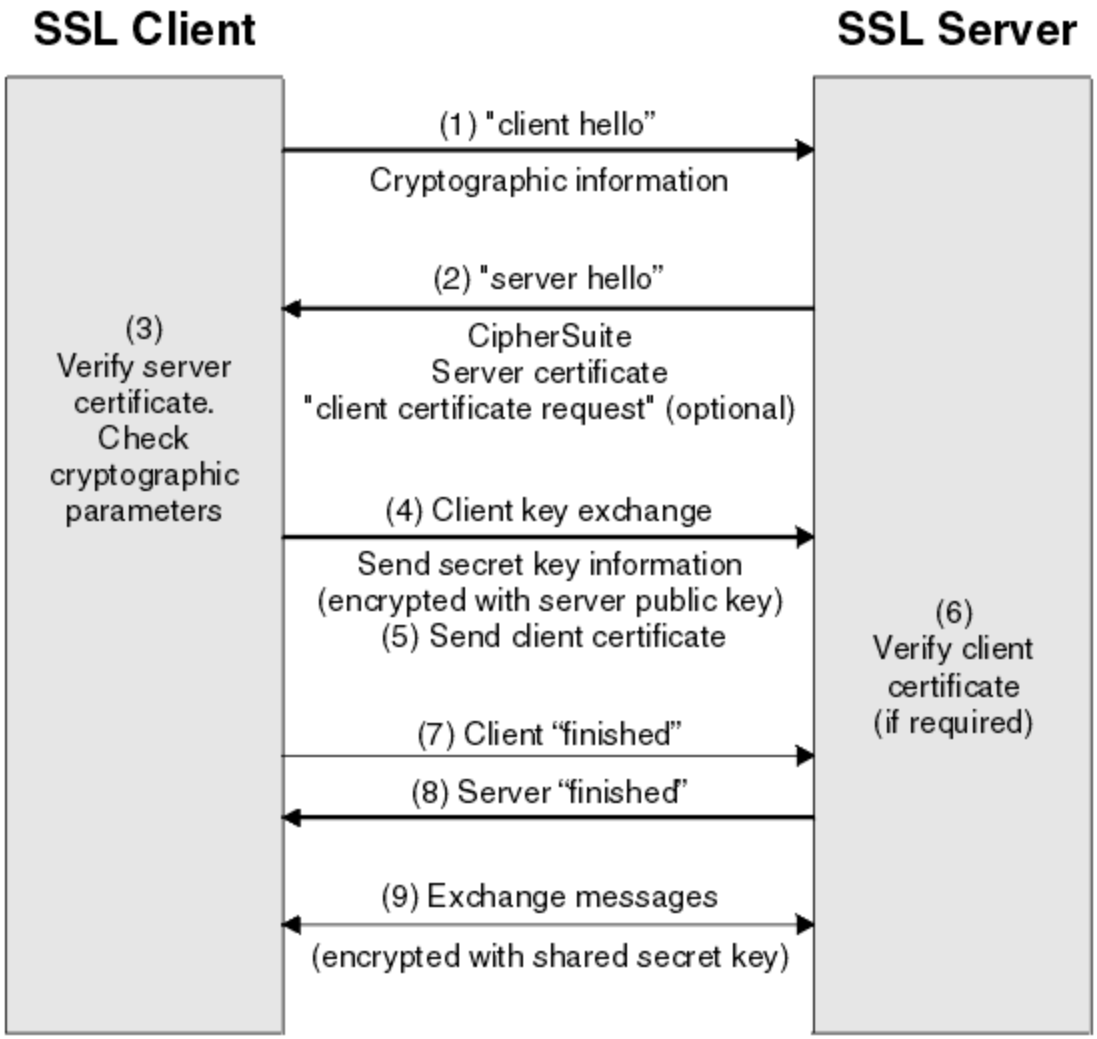
Forenzní analýza

- Nevypínat stroj!
- Pokud je možné, provádět minimum operací na aktivních filesystémech
- Sběr živých dat ze systému – otevřená spojení, data v ramdiscích, …
  - Pokud možno nepoužívat příkazy z kompromitovaného stroje
- Obraz disku a jeho následná off-line analýza
  - Existují specializované nástroje (coroner‘s toolkit, sleuth kit, …)
- Obnova stroje a vrácení do provozu
  - Report o incidentu!

Techniky útoků

- Útoky na protokol
  - SSL negotiations, WEP (RC4), MD5
- Chyba v implementaci
  - Neověřování vstupu (SQL injections)
  - XSS
  - Buffer overflows
  - openssl RND
- Sociální inženýrství
  - Phishing, Pharming
  - Hádání hesel

Útoky na protokoly - SSL



Útoky na protokoly - SSL

- SSL renegotiation
- Často používaná u webových serverů
  - Autentizace klientským certifikátem
  - Změna šifrovacího protokolu
  - Uživatelem iniciovaná renegociace
- Útočník
  - MITM útok
  - Přehrává komunikaci mezi klientem a SSL serverem
  - Po renegociaci útočník vloží svůj

Útoky na protokoly - WEP

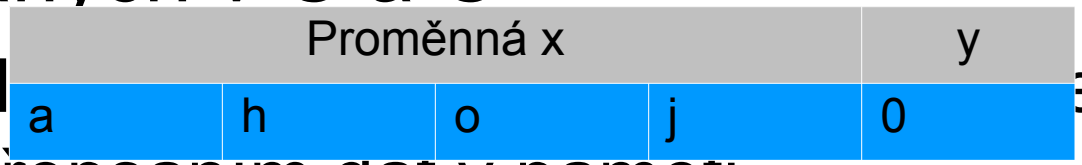
- Šifra RC4
  - Používá XOR nad CipherText, IV a zprávou
  - IV je pouze 24-bitový
  - Vyžaduje vždy jiný IV pro každou zprávu
- WEP
  - IV je přenášen v plain-text
  - Jiný IV pro každý paket, tzn. pro každý 16777216-tý paket je použit stejný IV jako pro první.
  - Každý paket začíná známými daty
  - Zjištěny vazby mezi CiphetText a šifrovanou

MD5

- V současné době již slabý hashovací algoritmus
- Hasovací fce musí být bezkolizní (2 různé zprávy musí generovat různý hash)
- U MD5 se současným výpočetním výkonem lze nalézt kolizní texty
- Dva různé X.509 certifikáty, které mají stejný podpis
- RapidSSL - 200 Playstation 3 za 2 dny

Buffer Overflows

- Nejčastěji se setkáváme u programů psaných v C a C++
  - N…em a přepsáním dat v paměti



- Stack based
  - Změna dat uložených na zásobníku
  - Manipulace s proměnnými
  - Změna návratové adresy

Sociální inženýrství

- Tyto techniky nejsou zaměřeny na chyby v protokolech nebo programech, využivají neznalosti, oklamání, zmatení uživatele.

Phishing

- Podvodné získávání citlivých údajů od uživatelů
- Rozeslání mailů s odkazem na podvodnou stránku, která vypadá stejně jako důvěryhodná a uživateli známá (el. Bankovnictví, webmail)
- Často se využívají maily ve formátu HTML:

<a href="http://evilsite.org/index.html">http://mojebanka.cz/zmensiheslo.php</a

Pharming

- Změna mapování DNS jméno → IP
- Uživatel vidí správné doménové jméno v URL, ale ta je umístěna na jiném stroji.
- Nejčastěji jsou obětí koncové stroje:
  - Počítače (soubor hosts)
  - Domácí routery (DNS relay)

Hádání hesel

- Příhlašení k webovým službám, SSH, FTP, …
- Metody získávání hesel
  - Brute force
  - Slovníkové útoky
  - Sociální inženýrství
- Využívání vlastností protokolů
  - Návratové hodnoty pro platné/neplatné záznamy

(D)DOS

- DOS – Denial of Service
  - Omezení/zastavení funkčnosti služby náhlým zvýšením požadavků k vyřízení
  - TCP/SYN flood
  - Webové servery
- DDOS – Distributed Denial of Service
  - Zdroj útoku není jeden počítač, ale celá síť
  - Především Botnety

Bezpečnost na webu

Slides based on 1-day Web Application Security Tutorial given by Ken van Wyk, KRvW Associates at FIRST/TF-CSIRT meeting in January 2010

Bezpečnost na webu

- Open Web Application Security Project (OWASP)
- OWASP Top-10
  - Klasifikace nejhorších bezpečnostních problémů na webu
- OWASP WebGoat
  - Výuková aplikace umožňující seznámení se s chybami a jejich zneužitím

OWASP Top-10

1. Cross site scripting
2. Injection flaws
3. Malicious file execution
4. Insecure direct object reference
5. Cross site request forgery
6. Information leakage and improper error handling
7. Broken authentication and session management
8. Insecure crypto storage
9. Insecure comms
10. Failure to restrict URL access

The only web app security rule

- Nothing in an HTTP Request can be trusted
  - EVER!
  - No kidding

#1 Cross site scripting (“XSS”)

- Can occur whenever a user can enter data into a web app
  - Consider all the ways a user can get data to the app
- When data is represented in browser, “data” can be dangerous
  - Consider this user input

```
<script>
 alert(document.cookie)
</script>
```

- Where can it happen?
  - ANY data input
- Two forms of XSS
  - Stored XSS
  - Reflected XSS

Stored XSS

- Attacker inputs script data on web app
  - Forums, “Contact Us” pages are prime examples
  - All data input must be considered
- Victim accidentally views data
  - Forum message, user profile, database field
- Can be years later
  - Malicious payload lies patiently in wait
  - Victim can be anywhere

Reflected XSS

- Attacker inserts script data into web app
- App immediately “reflects” data back
  - Search engines prime example
  - “String not found”
- Generally combined with other delivery mechanisms
  - HTML formatted spam most likely
  - Image tags containing search string as HTML parameter
    - Consider width=0 height=0 IMG SRC

XSS

- Why is this #1?
  - Input validation problems are pervasive
  - Focus on functional spec
- Why is it such a big deal?
  - *Highly* powerful attack
- Anything the user can do, the attacker can do
  - Take over session
  - Install malware
  - Copy/steal sensitive data
- *Reflected (via spam email) attacks most common technique by phishers*

#2 Injections flaws

- Most common is SQL injection
  - Attacker taints input data with SQL statement
  - SQL passes to SQL interpreter and runs
  - Question: Isn't XSS really just HTML injection?
- Consider the following input to an HTML form (via POST or GET)
  - Form requests a variable called "username"
  - Attacker enters `' or 1=1 --`
  - What happens next?

Other injections dangers

- SQL injection is common but others exist
  - XML
  - LDAP
  - Command shell
  - Comma delimited files
  - Log files
- Context is everything
  - Must be shielded from presentation layer
- Input validation will set you free
  - Positive validation is vital

#3 Malicious file execution

- Can occur whenever a user can directly affect an interpreted system resource name
  - Generally in combination with sending input to command interpreter
- Consider an app that displays a user supplied filename via a system call
  - User enters as filename `file.txt; rm -rf / &`
  - What happens?

#4 Insecure direct object reference

- Another input validation issue
- Unchecked user input allowing an attacker to access an unintended resource
- Examples include
  - Files
  - Sensitive application functions
  - Consider
    - `www.victim.com/AddUser.jsp?userid=123`
  - What if attacker changes to “321”?

Object reference issues

- Map objects in server code
- Many web apps use presentation layer security to “hide” sensitive functions
  - This approach is doomed to failure
- Strive for a positive input validation whenever possible
  - Map exposed names to system objects on the server
  - Discard all others
  - Applies for other mentioned issues as well!
- OS-layer data access control and compartmentalization also highly useful

#5 Cross site request forgery (CSRF)

- Relatively new, but potentially disastrous
- Attacker sends an image request to victim
  - During an active session on vulnerable app
  - Request may include malicious parameters
  - Response may include session cookie
- Consider if the image request arrived via spam email
  - Emailer renders the HTML and retrieves all “images”
  - Occurs while web browser is open and logged into popular banking site

CSRF Issues

- What's the big deal?
  - <img src='…'> can be used to hide commands other than images
  - Session cookies often have long timeout periods
  - Can redirect commands elsewhere on local network
    - 192.168.1.1 is very likely your web-enabled ADSL/ router ;-)
    - `http://192.168.1.1/admin/doSomething.ctl?username=admin&passwd=admin`
- CSRF remediation
  - This requires a lot of new coding
  - Very few existing web apps are protected
  - Phishers beginning to actively use this technique

Příklady útoků

- Chuck Norris

Zdroje k dalšímu studiu

- http://www.owasp.org
- http://www.terena.org/activities/tf-csirt/
- http://www.first.org/
- http://www.securityfocus.com
- Wikipedie:-)