サンプル・レポート

セキュリティ アセスメント レポート

CROSS HEAD - TESTserver01 (192.168.xxx.xxx)

レポート詳細

スキャン完了日: 06-04-2017

作成者: クロス・ヘッド

目次

このドキュメントについて

このセキュリティレポートは、F-Secure Radar脆弱性マネジメントソリューションにより生成され、「CROSS HEAD - TESTserver01」のセキュリティアセスメント・分析の結果を含んでいます。アセスメントに関連するあらゆる情報、セキュリティの欠陥や脆弱性については、それに対する確固とした対応策とともに、レポートの中に記載されています。「CROSS HEAD - TESTserver01」から許可を受けた個人はこのドキュメントを見る権利があります。このドキュメントは機密情報を含んでいます。

情報セキュリティアセスメントについて

情報セキュリティアセスメントの目的は、任意の時点での、スキャンを許可されたコンポーネントのセキュリティレベルを測ることです。アセスメントは、対象のセキュリティレベルについて優れた知見をもたらすにすぎず、それを持って、情報セキュリティを保証する唯一のプロセスとして使用すべきではありません。

アセスメントの結果は、さもなければ見落としたであろう知見をもたらしますが、検証や設計段階で見ることができた、すべての弱点や脆弱性を必ずしも見つけることはできません。つまり、アセスメントはセキュリティ上の問題を明らかにしますが、それをもって欠陥や脆弱性が存在しないことを証明するわけではありません。

加えて、セキュリティの防衛技術と攻撃技術は常に進化しています。時には、以前はまったく知られてなかった、完全に新しいタイプの脆弱性が発見されています。その為、アセスメントの結果は時間とともにその有効性を失うため、重要なビジネス機能に対する定期的なアセスメントを実施することが求められます。

1. エグゼクティブ サマリー

1.1. 調査の背景

この調査の目的は、TESTserver01 (192.168.xxx.xxx)のセキュリティの分析にあります。加えて、プログラムまたは設定に誤りにより、攻撃者や悪意のあるユーザーがなんらかの行動をとりうるか否かといったチェックも実施しています。

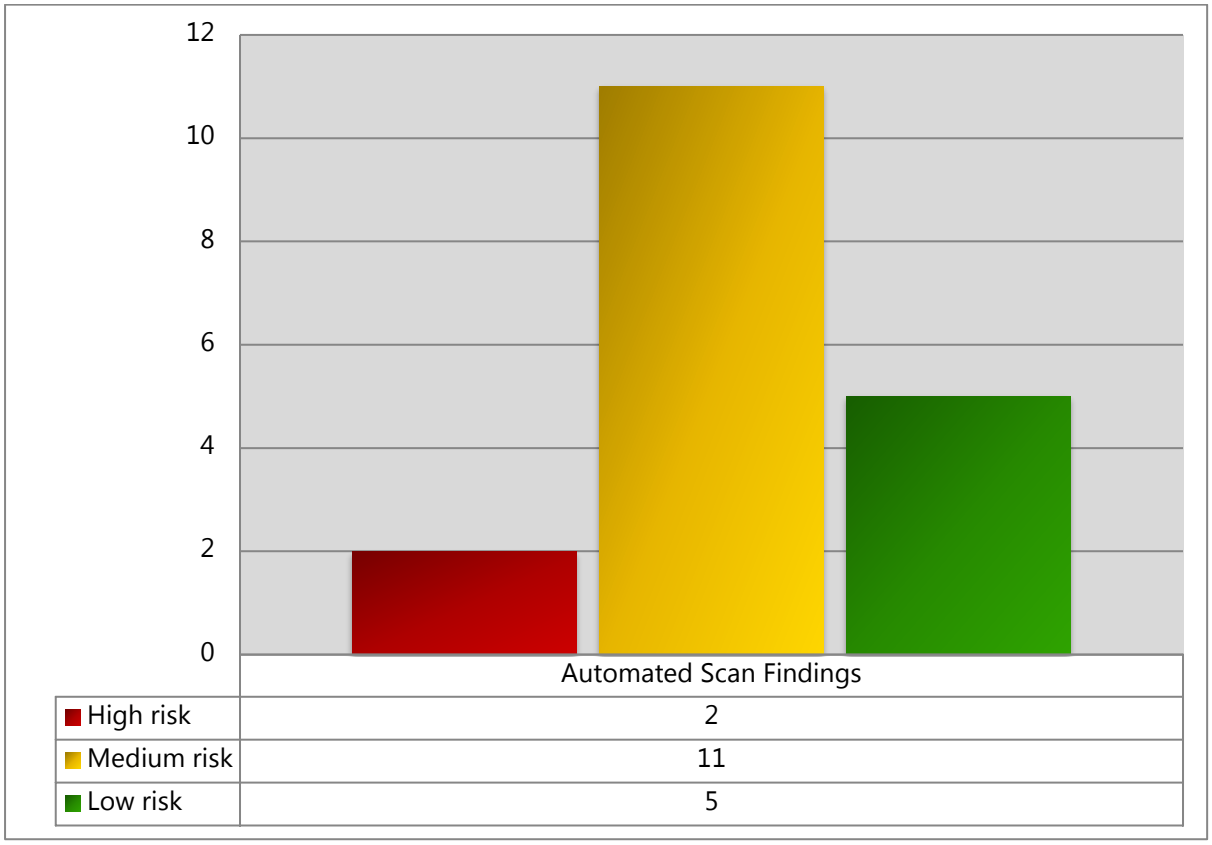
1.2. 結論

複数の脆弱性を発見したことにより、全体のセキュリティレベルは:**低** です。

1.3. 脆弱性の統計

セキュリティアセスメントを通じて発見されたすべての脆弱性をベースに、CVSSv2 メトリクスによる算定を行い、セキュリティランキングを生成します。

下記のグラフは、マネージャーがどの分野に注力すればよいのか、また至急対策を採らなければならないかどうかをあらわします。



検知内容(参考情報): 11

2. Findings – TESTserver01 (192.168.xxx.xxx)

Platform and services identified:

Target	Description
Name	TESTserver01 (192.168.xxx.xxx)
Platform	Linux (CentOS)
Service	Name: ssh Port: TCP/22 Banner: SSH-2.0-OpenSSH_5.3
Service	Name: udp Port: TCP/53
Service	Name: www Port: TCP/80 Banner: HTTP/1.0 400 Bad Request Server: squid/3.1.23 Mime-Version: 1.0 Date: Thu, 06 Apr 2017 02:33:46 GMT Content-Type: text/html Content-Length: 3133 X-Squid-Error: ERR_INVALID_URL 0 Vary: Accept-Language Content-Language: en X-Cache: MISS from proxy X-Cache-Lookup: NONE from proxy:8080 Via: 1.0 proxy (squid/3.1.23) Connection: close
Service	Name: (sunrpc) Port: TCP/111
Service	Name: (shell) Port: TCP/514
Service	Name: www Port: TCP/8080 Banner: HTTP/1.0 400 Bad Request Server: squid/3.1.23 Mime-Version: 1.0 Date: Thu, 06 Apr 2017 02:33:46 GMT Content-Type: text/html Content-Length: 3133 X-Squid-Error: ERR_INVALID_URL 0 Vary: Accept-Language Content-Language: en

	X-Cache: MISS from proxy X-Cache-Lookup: NONE from proxy:8080 Via: 1.0 proxy (squid/3.1.23) Connection: close
Service	Name: (med-fsp-rx) Port: TCP/24001
Service	Name: (unknown) Port: TCP/60557
Service	Name: dns Port: UDP/53
Service	Name: (sunrpc) Port: UDP/111

2.1. Automated scan results

2.1.1. High risk vulnerabilities

2.1.1.1 End-of-life product: Squid

High AV: **Network** AC: **High** Au: **None** C: **Complete** I: **Complete** A: **Complete** **7.6**

Vulnerability status: Unattended

Description

Squid proxy server has reached end-of-life status.

Active development for this version of Squid has ended. New updates or patches will not be available.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Migrate to the latest stable version of Squid 3.5.x.

2.1.1.2 Squid 3.x before 3.5.16 and 4.x before 4.0.8 Multiple Vulnerabilities

High AV: **Network** AC: **Low** Au: **None** C: **None** I: **Partial** A: **Partial** **7.5**

Vulnerability status: Unattended

Description

The remote proxy server is affected by multiple vulnerabilities.

Heap-based buffer overflow in the Icmp6::Recv function in icmp/Icmp6.cc in the pinger in Squid before 3.5.16 and 4.x before 4.0.8 allows remote servers to cause a denial of service (performance degradation or transition failures) or write sensitive information to log files via an ICMPv6 packet.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Upgrade to the latest version of the Squid proxy software. The vendor has prepared a fix for this issue in versions 3.5.16 and 4.0.8.

Tags

Squid, Version Based, Web Server

External references

CVE-2016-3947

2.1.2. Medium risk vulnerabilities

2.1.2.1 Squid 2.x, 3.x before 3.5.17 and 4.x before 4.0.9 Multiple Vulnerabilities

Medium AV: **Network** AC: **Medium** Au: **None** C: **Partial** I: **Partial** A: **Partial** **6.8**

Vulnerability status: Unattended

Description

The remote proxy server is affected by multiple vulnerabilities.

Buffer overflow in cachemgr.cgi in Squid 2.x, 3.x before 3.5.17, and 4.x before 4.0.9 might allow remote attackers to cause a denial of service or execute arbitrary code by seeding manager reports with crafted data.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Upgrade to the latest version of the Squid proxy software. The vendor has prepared a fix for this issue in versions 3.5.17 and 4.0.9.

Tags

Squid, Version Based, Web Server

External references

CVE-2016-4051

2.1.2.2 Squid 3.x before 3.5.17 and 4.x before 4.0.9 Multiple Vulnerabilities

Medium AV: **Network** AC: **Medium** Au: **None** C: **Partial** I: **Partial** A: **Partial** **6.8**

Vulnerability status: Unattended

Description

The remote proxy server is affected by multiple vulnerabilities.

[CVE-2016-4052] Multiple stack-based buffer overflows in Squid 3.x before 3.5.17 and 4.x before 4.0.9 allow remote HTTP servers to cause a denial of service or execute arbitrary code via crafted Edge Side Includes (ESI) responses.

[CVE-2016-4053] Squid 3.x before 3.5.17 and 4.x before 4.0.9 allow remote attackers to obtain sensitive stack layout information via crafted Edge Side Includes (ESI) responses, related to incorrect use of assert and compiler optimization.

[CVE-2016-4054] Buffer overflow in Squid 3.x before 3.5.17 and 4.x before 4.0.9 allows remote attackers to execute arbitrary code via crafted Edge Side Includes (ESI) responses.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Upgrade to the latest version of the Squid proxy software. The vendor has prepared a fix for this issue in versions 3.5.17 and 4.0.9.

Tags

Squid, Version Based, Web Server

External references

CVE-2016-4052

CVE-2016-4053

CVE-2016-4054

2.1.2.3 Squid before 3.5.6 CONNECT Method Handler Privilege Escalation Vulnerability

Medium AV: **Network** AC: **Medium** Au: **None** C: **Partial** I: **Partial** A: **Partial** **6.8**

Vulnerability status: Unattended

Description

The remote proxy server is affected by a privilege escalation vulnerability.

Squid before 3.5.6 does not properly handle CONNECT method peer responses when configured with cache_peer, which allows remote attackers to bypass intended restrictions and gain access to a backend proxy via a CONNECT request.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Upgrade to the latest version of the Squid proxy software. The vendor has prepared a fix for this issue in version 3.5.6.

Tags

Squid, Version Based, Web Server

External references

CVE-2015-5400

2.1.2.4 HTTP TRACE method allowed

Medium AV: **Network** AC: **Medium** Au: **None** C: **Partial** I: **Partial** A: **None** **5.8**

Vulnerability status: Unattended

Description

The remote web server has the TRACE method enabled.

The remote web server supports the TRACE method. TRACE is a HTTP method that is used to debug web server connections. This method should not be enabled on production servers as it can be exploited to conduct cross-site scripting against users of the website.

The vulnerability is based on the following retrieved information from 80/TCP:

```
HTTP TRACE method enabled:

Request:

TRACE /ssng/test123.html HTTP/1.1
Host: 192.168.xxx.xxx
User-Agent: Mozilla/4.0 (Mozilla/4.0; MSIE 7.0; Windows NT 5.1; FDM; SV1; .NET CLR
3.0.04506.30) F-Secure Radar
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.8
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Cache-Control: no-cache
Connection: Close


Response:

HTTP/1.1 200 OK
Date: Thu, 06 Apr 2017 03:11:11 GMT
Server: Apache/2.2.15 (CentOS)
Connection: close
Transfer-Encoding: chunked
Content-Type: message/http

TRACE /ssng/test123.html HTTP/1.1
Host: 192.168.xxx.xxx
User-Agent: Mozilla/4.0 (Mozilla/4.0; MSIE 7.0; Windows NT 5.1; FDM; SV1; .NET CLR
3.0.04506.30) F-Secure Radar
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.8
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Cache-Control: no-cache
Connection: Close
```

Recommendations

Disable the HTTP TRACE method.

For Apache (with mod_rewrite):

Disable the TRACE method by enabling mod_rewrite and setting the following directives in httpd.conf:

RewriteEngine On

RewriteCond %{REQUEST_METHOD} ^TRACE

RewriteRule .*

- [F]

For Apache (without mod_rewrite):

Set the following Apache directive to 'TraceEnable off' in the main server config.

For IIS:

Setup Microsoft URL Scan and configure the urlscan.ini as follows. Set the 'UseAllowVerbs' parameter to 1 and specify only GET, HEAD, and POST in the [AllowVerbs] section.

In order to verify this vulnerability, issue either a 'TRACE / HTTP/1.0' request to the host using e.g. telnet.

Tags

Web Server

External references

CVE-2004-2320

CVE-2007-3008

CVE-2010-0386

2.1.2.5 Directory listings enabled

Medium AV: **Network** AC: **Low** Au: **None** C: **Partial** I: **None** A: **None** **5.0**

Vulnerability status: Unattended

Description

Directory listings enable an attacker to learn details about the framework in use.

Directory listings enable an attacker to learn details about the framework in use, like exact version number and information about installed components.

An attacker can gain information about the web application by browsing directory listings that reveal files and folder hierarchy in the application. This information can be used to exploit vulnerabilities in the web application.

Any sensitive resources within your web root should be properly access-controlled and should not be accessible to an unauthorized party who knows the URL. Nevertheless, directory listings can aid an attacker by enabling them to quickly identify the resources at a given path, and proceed directly to analyzing and attacking them.

The vulnerability is based on the following retrieved information from 80/TCP:

```
http://192.168.xxx.xxx:80/icons/small/
```

Recommendations

There is not usually any good reason to provide directory listings, and disabling them may place additional hurdles in the path of an attacker. This can normally be achieved in two ways:

- Configure your web server to prevent directory listings for all paths beneath the web root;

- Place into each directory a default file (such as index.htm) which the web server will display instead of returning a directory listing.

Tags

Web Server

2.1.2.6 Squid 3.5 before 3.5.23 and 4.0 before 4.0.17 Information Disclosure Vulnerability

Medium AV: **Network** AC: **Low** Au: **None** C: **Partial** I: **None** A: **None** **5.0**

Vulnerability status: Unattended

Description

The remote proxy server is affected by an information disclosure vulnerability.

Incorrect HTTP Request header comparison in Squid HTTP Proxy 3.5.0.1 through 3.5.22, and 4.0.1 through 4.0.16 results in Collapsed Forwarding feature mistakenly identifying some private responses as being suitable for delivery to multiple clients.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Upgrade to the latest version of the Squid proxy software. The vendor has prepared a fix for this issue in versions 3.5.23 and 4.0.17.

Tags

Squid, Version Based, Web Server

External references

CVE-2016-10003

http://www.securityfocus.com/bid/94953

2.1.2.7 Squid 3.x before 3.5.15 and 4.x before 4.0.7 Denial of Service Vulnerability

Medium AV: **Network** AC: **Low** Au: **None** C: **None** I: **None** A: **Partial** **5.0**

Vulnerability status: Unattended

Description

The remote proxy server is affected by a denial of service vulnerability.

[CVE-2016-2570] The Edge Side Includes (ESI) parser in Squid 3.x before 3.5.15 and 4.x before 4.0.7 does not check buffer limits during XML parsing, which allows remote HTTP servers to cause a denial of service (assertion failure and daemon exit) via a crafted XML document, related to esi/CustomParser.cc and esi/CustomParser.h.

[CVE-2016-2571] http.cc in Squid 3.x before 3.5.15 and 4.x before 4.0.7 proceeds with the storage of certain data after a response-parsing failure, which allows remote HTTP servers to cause a denial of service (assertion failure and daemon exit) via a malformed response.

[CVE-2016-2569] Squid 3.x before 3.5.15 and 4.x before 4.0.7 does not properly append data to String objects, which allows remote servers to cause a denial of service (assertion failure and daemon exit) via a long string, as demonstrated by a crafted HTTP Vary header.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Upgrade to the latest version of the Squid proxy software. The vendor has prepared a fix for this issue in versions 3.5.15 and 4.0.7.

Tags

Squid, Version Based, Web Server

External references

CVE-2016-2569

CVE-2016-2570

CVE-2016-2571

2.1.2.8 Squid 3.x before 3.5.16 and 4.x before 4.0.8 Denial of Service Vulnerability

Medium AV: **Network** AC: **Low** Au: **None** C: **None** I: **None** A: **Partial** **5.0**

Vulnerability status: Unattended

Description

The remote proxy server is affected by a denial of service vulnerability.

Squid 3.x before 3.5.16 and 4.x before 4.0.8 improperly perform bounds checking, which allows remote attackers to cause a denial of service via a crafted HTTP response, related to Vary headers.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Upgrade to the latest version of the Squid proxy software. The vendor has prepared a fix for this issue in versions 3.5.16 and 4.0.8.

Tags

Squid, Version Based, Web Server

External references

CVE-2016-3948

2.1.2.9 Squid 3.x before 3.5.18 and 4.x before 4.0.10 Denial of Service Vulnerability

Medium AV: **Network** AC: **Low** Au: **None** C: **None** I: **None** A: **Partial** **5.0**

Vulnerability status: Unattended

Description

The remote proxy server is affected by a denial of service vulnerability.

[CVE-2016-4556] Double free vulnerability in Esi.cc in Squid 3.x before 3.5.18 and 4.x before 4.0.10 allows remote servers to cause a denial of service (crash) via a crafted Edge Side Includes (ESI) response.

[CVE-2016-4555] client_side_request.cc in Squid 3.x before 3.5.18 and 4.x before 4.0.10 allows remote servers to cause a denial of service (crash) via crafted Edge Side Includes (ESI) responses.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Upgrade to the latest version of the Squid proxy software. The vendor has prepared a fix for this issue in versions 3.5.18 and 4.0.10.

Tags

Squid, Version Based, Web Server

External references

CVE-2016-4555

CVE-2016-4556

2.1.2.10 Squid before 3.5.23 and 4.0.17 Information Disclosure Vulnerability

Medium AV: **Network** AC: **Low** Au: **None** C: **Partial** I: **None** A: **None** **5.0**

Vulnerability status: Unattended

Description

The remote proxy server is affected by an information disclosure vulnerability.

Incorrect processing of responses to If-None-Modified HTTP conditional requests in Squid HTTP Proxy 3.1.10 through 3.1.23, 3.2.0.3 through 3.5.22, and 4.0.1 through 4.0.16 leads to client-specific Cookie data being leaked to other clients. Attack requests can easily be crafted by a client to probe a cache for this information.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
squid/3.1.23
```

Recommendations

Upgrade to the latest version of the Squid proxy software. The vendor has prepared a fix for this issue in versions 3.5.23 and 4.0.17.

Tags

Squid, Version Based, Web Server

External references

CVE-2016-10002

http://www.securityfocus.com/bid/94953

2.1.2.11 SSH Server Supports RC4

Medium AV: **Network** AC: **Medium** Au: **None** C: **Partial** I: **None** A: **None** **4.3**

Vulnerability status: Unattended

Description

The remtoe ssh server offers weak encryption.

RC4 (arcfour) as used in SSH is known to suffer from a number of weaknesses and is considered as weak by a modern standards.

The vulnerability is based on the following retrieved information from 22/TCP:

```
RC4 ciphers offered by the server:
arcfour256,
arcfour128,
arcfour
```

Recommendations

Disable RC4 cipher in your ssh server configuration.

2.1.3. Low risk vulnerabilities

2.1.3.1 Missing HTTP security headers

Low AV: **Network** AC: **High** Au: **None** C: **None** I: **Partial** A: **None** **2.6**

Vulnerability status: Unattended

Description

The remote web server is missing optional HTTP security headers

The website's security posture can be enhanced by defining several HTTP headers designed for improving end user security. As today's attacks increasingly target the client, the use of security enhancing browser features is encouraged.

Strict-Transport-Security

HTTP Strict Transport Security header instructs the browser to access the site using only HTTPS connections. The header mitigates the effects of man-in-the-middle attacks against end users. Even though the header is not yet supported by all browser vendors, it is likely to be included in future versions of popular browsers.

X-Content-Type-Options: nosniff

Internet Explorer has historically had MIME-type detection features which enable an attacker to execute JavaScript from a plaintext file. The above header instructs the browser to strictly follow the MIME-type defined in the Content header, preventing XSS attacks in certain attack scenarios where user uploaded documents are served.

X-XSS-Protection 1; mode=block

Some modern browsers such as IE8 and Google Chrome contain an XSS filter which tries to prevent exploitation of reflected cross site scripting vulnerabilities. Websites can explicitly define the browser to block such attacks. The block mode instructs the browser to block the whole page instead of modifying the server response if an attack is detected.

X-Frame-Options: DENY

If a website allows its content to be presented within a third party frame, an attacker can perform so-called clickjacking attacks which are similar to Cross-Site Request Forgery. These attacks can be prevented with adding an X-Frame-Options header which instructs the browser not to render the website within a frame.

Content-Security-Policy

Is a declarative policy that lets the authors (or server administrators) of a web application inform the client about the sources from which the application expects to load resources. To

mitigate XSS attacks, for example, a web application can declare that it only expects to load script from specific, trusted sources. This declaration allows the client to detect and block malicious scripts injected into the application by an attacker. Content Security Policy is not intended as a first line of defence against content injection vulnerabilities. Instead, CSP is best used as defence-in-depth, to reduce the harm caused by content injection attacks.

The vulnerability is based on the following retrieved information from 8080/TCP:

```
The following HTTP headers are missing:
X-Content-Type-Options
X-XSS-Protection
X-Frame-Options
Content-Security-Policy
```

Recommendations

Perform cost/benefit analysis for implementing the listed HTTP security headers.

Tags

Web Server

2.1.3.2 Missing HTTP security headers

Low AV: **Network** AC: **High** Au: **None** C: **None** I: **Partial** A: **None** **2.6**

Vulnerability status: Unattended

Description

The remote web server is missing optional HTTP security headers

The website's security posture can be enhanced by defining several HTTP headers designed for improving end user security. As today's attacks increasingly target the client, the use of security enhancing browser features is encouraged.

Strict-Transport-Security

HTTP Strict Transport Security header instructs the browser to access the site using only HTTPS connections. The header mitigates the effects of man-in-the-middle attacks against end users. Even though the header is not yet supported by all browser vendors, it is likely to be included in future versions of popular browsers.

X-Content-Type-Options: nosniff

Internet Explorer has historically had MIME-type detection features which enable an attacker to execute JavaScript from a plaintext file. The above header instructs the browser to strictly follow the MIME-type defined in the Content header, preventing XSS attacks in certain attack scenarios where user uploaded documents are served.

X-XSS-Protection 1; mode=block

Some modern browsers such as IE8 and Google Chrome contain an XSS filter which tries to prevent exploitation of reflected cross site scripting vulnerabilities. Websites can explicitly define the browser to block such attacks. The block mode instructs the browser to block the whole page instead of modifying the server response if an attack is detected.

X-Frame-Options: DENY

If a website allows its content to be presented within a third party frame, an attacker can perform so-called clickjacking attacks which are similar to Cross-Site Request Forgery. These attacks can be prevented with adding an X-Frame-Options header which instructs the browser not to render the website within a frame.

Content-Security-Policy

Is a declarative policy that lets the authors (or server administrators) of a web application inform the client about the sources from which the application expects to load resources. To mitigate XSS attacks, for example, a web application can declare that it only expects to load script from specific, trusted sources. This declaration allows the client to detect and block

malicious scripts injected into the application by an attacker. Content Security Policy is not intended as a first line of defence against content injection vulnerabilities. Instead, CSP is best used as defence-in-depth, to reduce the harm caused by content injection attacks.

The vulnerability is based on the following retrieved information from 80/TCP:

```
The following HTTP headers are missing:
X-Content-Type-Options
X-XSS-Protection
X-Frame-Options
Content-Security-Policy
```

Recommendations

Perform cost/benefit analysis for implementing the listed HTTP security headers.

Tags

Web Server

2.1.3.3 SSH Downgrade Attack (SLOTH)

Low AV: **Network** AC: **High** Au: **None** C: **Partial** I: **None** A: **None** **2.6**

Vulnerability status: Unattended

Description

The remote server is vulnerable to a downgrade attack.

An attacker can downgrade the negotiated ciphersuite to a weak algorithm using the chosen-prefix transcript collision attack on the SHA1 function as used in Diffie-Hellman exchange phase of SSH connection.

The vulnerability is based on the following retrieved information from 22/TCP:

```
Vulnerable KEXs offered by the server:
diffie-hellman-group-exchange-sha1,
diffie-hellman-group14-sha1,
diffie-hellman-group1-sha1
```

Recommendations

Disable key exchange algorithms using SHA1 in your ssh server configuration.

2.1.3.4 SSH Insecure Diffie-Hellman Key Exchange Configuration

Low AV: **Network** AC: **High** Au: **None** C: **Partial** I: **None** A: **None** **2.6**

Vulnerability status: Unattended

Description

The remote SSH server uses insecure key exchange algorithms configuration.

SSH handshakes use Diffie-Hellman algorithm for the key exchange. Performing precomputations on a fixed/standarized groups of integers would allow a passive eavesdropper to decrypt the traffic. An attack on a single 512-bit has been proven. For the 768-bit and 1024-bit groups it is believed that computations are plausible given appropriate resources. For example, the diffie-hellman-group1-sha1 mechanism available in OpenSSH which uses the fixed 1024-bit Oakley Group 2 could be broken by the nation-state attackers.

The vulnerability is based on the following retrieved information from 22/TCP:

```
diffie-hellman-group1-sha1 key exchange algorithm is supported by the server
```

Recommendations

Configure your SSH server so it uses moduli longer than 1024 bits and make sure that the diffie-hellman-group1-sha1 algorithm is disabled.

2.1.3.5 SSH Server Supports Weak Ciphers

Low AV: **Network** AC: **High** Au: **None** C: **Partial** I: **None** A: **None** **2.6**

Vulnerability status: Unattended

Description

The remote SSH server offers weak encryption.

If weak ciphers are used by SSH to protect the session data, it is possible for a third party to record the network traffic, mount an offline bruteforcing attack, recover the session key and from there recover the content of the whole SSH session. It is perhaps also possible to recover usernames, passwords and other sensitive information.

3DES, BLOWFISH and CAST ciphers have 64-bit block size, and they are advised only for legacy use. Cipher algorithms in Cipher Block Chaining (CBC) mode suffer from the padding oracle attacks and are considered unsafe.

The vulnerability is based on the following retrieved information from 22/TCP:

```
Weak ciphers offered by the server:
aes128-cbc,
3des-cbc,
blowfish-cbc,
cast128-cbc,
aes192-cbc,
aes256-cbc
```

Recommendations

Disable weak ciphers in your ssh server configuration.

2.1.4. Informational findings

2.1.4.1 Backported Software Detected

Vulnerability status: Unattended

Description

The remote host is using backporting.

Backporting is a process of patching an older version of software rather than upgrading to a newer version. Vulnerability checks that are basing on the version number may produce false positives as they do not take into account backported security fixes. In order to avoid false positives these checks are going to be disabled. This behavior can be disabled in the scan settings.

The informational finding is based on the following retrieved information from 0/TCP:

```
Detected operating system (CentOS) is known to use backporting.
```

Recommendations

This finding is informational, no action is required.

Tags

Informational, Service detection

2.1.4.2 ICMP Address Mask and/or Timestamp Requests Allowed From Arbitrary Hosts

Vulnerability status: Unattended

Description

Remote server responds to ICMP timestamp and/or address mask requests.

ICMP information such as (1) netmask and (2) timestamp is allowed from arbitrary hosts. This method of attack provides information that could help an attacker to identify other vulnerabilities, but does no direct harm.

The informational finding is based on the following retrieved information from 0/ICMP:

```
Remote host responds to ICMP Timestamp Request
```

Recommendations

Configure your firewall to block ICMP packets of type 13/17 or disable ICMP replies for type 13/17 on your system.

Tags

Informational

External references

CVE-1999-0524

2.1.4.3 Ping the remote host

Vulnerability status: Unattended

Description

This plugin checks if the remote host responses to ping.

This plugin checks if the remote host is alive using one or more ping types:

- An ARP ping, if the host is on the local subnet

- An ICMP ping

- A TCP ping, which sends a packet with the flag SYN, and the host will reply with a RST or a SYN/ACK

- A UDP ping (DNS, RPC, NTP, etc.)

The informational finding is based on the following retrieved information from 0/ICMP:

```
The remote host replied to an ICMP ping packet
```

Recommendations

This finding is informational, no action is required.

Tags

Informational

2.1.4.4 robots.txt was not found

Vulnerability status: Unattended

Description

The robots.txt file was not found on the remote host.

The robots.txt file is used to give instructions to web robots also known as search engines such as Google, Bing and others. The file defines what content web robots are allowed to index, which robots you allow and much more.

When robots.txt does not exist, web robots will attempt to index the web application. In that case, ensure that the web application do not expose any sensitive information.

For more information please see the following references:

- List of web robots http://www.robotstxt.org/db.html

- Paper about robots.txt http://www.sans.org/reading_room/whitepapers/awareness/robotstxt_33955

The informational finding is based on the following retrieved information from 80/TCP:

```
The robots.txt file is not available on the remote host
```

Recommendations

Define rules for web robots and add robots.txt file.

Tags

Informational

2.1.4.5 RPC portmapper Service Detection

Vulnerability status: Unattended

Description

An ONC RPC portmapper is running on the remote host.

The RPC portmapper is running on this port. The portmapper allows someone to get the port number of each RPC service running on the remote host by sending either multiple lookup requests or a DUMP request.

Recommendations

This finding is informational, no action is required.

Tags

Informational

External references

CVE-1999-0632

2.1.4.6 Service Detection: DNS

Vulnerability status: Unattended

Description

DNS service is running on the remote host.

DNS (Domain Name System) is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network.

The informational finding is based on the following retrieved information from 53/UDP:

```
DNS (Domain Name System) service has been detected on this port
```

Recommendations

This finding is informational, no action is required.

Tags

Informational, Service detection

2.1.4.7 Service Detection: Postgres

Vulnerability status: Unattended

Description

Postgres, is an object-relational database management system.

Postgres, is an object-relational database management system.

The informational finding is based on the following retrieved information from 24001/TCP:

```
PostgreSQL Posible versions: 9.1.10, 9.1.11, 9.1.5, 9.1.6, 9.1.7, 9.1.8, 9.1.9 is
running on the remote host.
```

Recommendations

This finding is informational, no action is required.

Tags

Informational, Service detection

2.1.4.8 Service Detection: SSH

Vulnerability status: Unattended

Description

An SSH server is running on the remote host.

SSH (Secure Shell) is a network protocol for initiating secure shell sessions on remote machines.

The informational finding is based on the following retrieved information from 22/TCP:

```
An SSH server is running on this port

Banner:

SSH-2.0-OpenSSH_5.3
```

Recommendations

This finding is informational, no action is required.

Tags

Informational, Service detection, SSH

2.1.4.9 Service Detection: WWW

Vulnerability status: Unattended

Description

An HTTP server is running on the remote host.

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.

The informational finding is based on the following retrieved information from 8080/TCP:

```
An HTTP server is running on this port

Banner:

squid/3.1.23
```

Recommendations

This finding is informational, no action is required.

Tags

Informational, Service detection, Web Server

2.1.4.10 Service Detection: WWW (Apache HTTP Server)

Vulnerability status: Unattended

Description

The remote host is running Apache HTTP Server.

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. The Apache HTTP Server is a project of The Apache Software Foundation.

The informational finding is based on the following retrieved information from 80/TCP:

```
Apache HTTP Server is running on this port.
Banner:
HTTP/1.1 200 OK
Date: Thu, 06 Apr 2017 02:29:39 GMT
Server: Apache/2.2.15 (CentOS)
Last-Modified: Wed, 10 Feb 2016 08:23:59 GMT
ETag: "20182-251-52b6627e5d9d7"
Accept-Ranges: bytes
Content-Length: 593
Connection: close
Content-Type: text/html; charset=UTF-8
```

Recommendations

This finding is informational, no action is required.

Tags

Apache, Informational, Service detection, Web Server

2.1.4.11 SSH Server Configuration

Vulnerability status: Unattended

Description

The remote SSH server uses the following configuration.

ServerHostKeyAlgorithms specifies the host key algorithms that the server offers.

ServerKeyExchangeAlgorithms specifies the available KEX (Key Exchange) algorithms.

ServerEncryptionAlgorithms specifies supported symmetric ciphers.

ServerMacAlgorithms specifies the available MAC (message authentication code) algorithms used for data integrity protection.

ServerAllowedAuthentications specifies the available authentication methods for a user to be granted access.

The informational finding is based on the following retrieved information from 22/TCP:

```
ServerHostKeyAlgorithms:
ssh-rsa,
ssh-dss

ServerKeyExchangeAlgorithms:
diffie-hellman-group-exchange-sha256,
diffie-hellman-group-exchange-sha1,
diffie-hellman-group14-sha1,
diffie-hellman-group1-sha1

ServerEncryptionAlgorithms:
aes128-ctr,
aes192-ctr,
aes256-ctr,
arcfour256,
arcfour128,
aes128-cbc,
3des-cbc,
blowfish-cbc,
cast128-cbc,
aes192-cbc,
aes256-cbc,
arcfour,
rijndael-cbc@lysator.liu.se

ServerMacAlgorithms:
hmac-md5,
hmac-sha1,
umac-64@openssh.com,
hmac-sha2-256,
hmac-sha2-512,
hmac-ripemd160,
hmac-ripemd160@openssh.com,
hmac-sha1-96,
hmac-md5-96
```

```
ServerAllowedAuthentications:
publickey,
gssapi-keyex,
gssapi-with-mic,
password
```

Recommendations

This finding is informational, no action is required.

Tags

Informational, Service detection

3. APPENDIX

3.1. About test methodology

The complexity of expanded infrastructures and modern IT solutions triggers the need of complete, deep and well-balanced security assessments. To face this challenge, F-Secure has developed a proprietary methodology aiming to evaluate the security of the requested environments.

In general, the vulnerability scanning process consists of four phases.

3.1.1. Reconnaissance

This phase gives a very general overview of the target environment. The goal is to figure out what kind of components and services are present in the infrastructure. The information about the targets (hosts, URLs, credentials) may be collected by various means such as, WHOIS databases, and DNS including white intelligence techniques as well as the input given by the customer. Moreover, network discovery using F-Secure's proprietary network scanner, F-Secure Radar Discovery Scan, is performed in order to identify systems present in customer's networks.

3.1.2. Enumeration

During this phase, the information gathered in the previous step is utilized to steer the more detailed scanning against individual components and services. The platform scan is performed with F-Secure Radar System Scan. The web applications are scanned using the F-Secure Radar Web Scanner.

3.1.3. Research for vulnerabilities and (optional) exploitation

This is an optional and manual phase where the F-Secure Radar user put in most of the effort. The vulnerabilities identified in the previous phase need to be thoroughly verified to avoid (as much as it is possible) false positives and provide appropriate quality of the results.

3.1.4. Reporting

The final phase is reporting, the process of documenting all of the vulnerabilities identified during the assessment. Every finding is documented with a detailed description including the exact location, conditions when the vulnerability occurs (together with information allowing the customer to reproduce the finding), security impact analysis, and proposed remediation strategy. The goal is to provide both detailed and precise information about security issues, but also to suggest the best way to mitigate them.

F-Secure Radar has a powerful reporting engine that allows the end-user to customize the content of the report and download reports in various formats.

3.2. About CVSS scoring

The findings are scored using the international CVSSv2 metrics. The goal of the scoring system is to find common metrics for the findings. Using a common scoring system allows comparison of findings between assignments. It is worth noting however, that the numerical value (CVSSv2 score) is only meant for general guidance and should be interpreted as such.

3.2.1. About base metrics

The base metric group captures the characteristics of a vulnerability that are constant with time and across user environments. The Access Vector, Access Complexity, and Authentication metrics capture how the vulnerability is accessed and whether or not extra conditions are required to exploit it. The three impact metrics measure how a vulnerability, if exploited, will directly affect an IT asset, where the impacts are independently defined as the degree of loss of confidentiality, integrity, and availability. For example, a vulnerability could cause a partial loss of integrity and availability, but no loss of confidentiality.

3.2.2. Access Vector (AV)

This metric reflects how the vulnerability is exploited. The possible values for this metric are listed in the table below. The more remote an attacker can be to attack a host, the greater the vulnerability score.

3.2.2.1 Local (L)

A vulnerability exploitable with only *local access* requires the attacker to have either physical access to the vulnerable system or a local (shell) account. Examples of locally exploitable vulnerabilities are peripheral attacks such as Firewire/USB DMA attacks, and local privilege escalations (e.g., sudo).

3.2.2.2 Adjacent Network (A)

A vulnerability exploitable with *adjacent network access* requires the attacker to have access to either the broadcast or collision domain of the vulnerable software. Examples of local networks include local IP subnet, Bluetooth, IEEE 802.11, and local Ethernet segment.

3.2.2.3 Network (N)

A vulnerability exploitable with *network access* means the vulnerable software is bound to the network stack and the attacker does not require local network access or local access. Such a vulnerability is often termed "remotely exploitable". An example of a network attack is an RPC buffer overflow.

3.2.3. Access Complexity (AC)

This metric measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. For example, consider a buffer overflow in an Internet service: once the target system is located, the attacker can launch an exploit at will.

3.2.3.1 High (H)

Specialized access conditions exist. For example:

- In most configurations, the attacking party must already have elevated privileges or spoof additional systems in addition to the attacking system (e.g., DNS hijacking).
- The attack depends on social engineering methods that would be easily detected by knowledgeable people. For example, the victim must perform several suspicious or atypical actions.
- The vulnerable configuration is seen very rarely in practice.
- If a race condition exists, the window is very narrow.

3.2.3.2 Medium (M)

The access conditions are somewhat specialized; the following are examples:

- The attacking party is limited to a group of systems or users at some level of authorization, possibly untrusted.
- Some information must be gathered before a successful attack can be launched.
- The affected configuration is non-default, and is not commonly configured (e.g., a vulnerability present when a server performs user account authentication via a specific scheme, but not present for another authentication scheme).
- The attack requires a small amount of social engineering that might occasionally fool cautious users (e.g., phishing attacks that modify a web browsers status bar to show a false link, having to be on someones buddy list before sending an IM exploit).

3.2.3.3 Low (L)

Specialized access conditions do not exist. The following are examples:

- The affected product typically requires access to a wide range of systems and users, possibly anonymous and untrusted (e.g., Internet-facing web or mail server).
- The affected configuration is default or ubiquitous.
- The attack can be performed manually and requires little skill or additional information gathering.
- The race condition is a lazy one (i.e., it is technically a race but easily winnable).

3.2.4. Authentication (Au)

This metric measures the number of times an attacker must authenticate to a target in order to exploit a vulnerability. This metric does not gauge the strength or complexity of the authentication process, only that an attacker is required to provide credentials before an exploit may occur. The possible values for this metric are listed in Table 3. The fewer authentication instances that are required, the higher the vulnerability score.

3.2.4.1 Multiple (M)

Exploiting the vulnerability requires that the attacker authenticate two or more times, even if the same credentials are used each time. An example is an attacker authenticating to an operating system in addition to providing credentials to access an application hosted on that system.

3.2.4.2 Single (S)

The vulnerability requires an attacker to be logged into the system (such as at a command line or via a desktop session or web interface).

3.2.4.3 None (N)

Authentication is not required to exploit the vulnerability.

3.2.5. Confidentiality Impact (C)

This metric measures the impact on confidentiality of a successfully exploited vulnerability. Confidentiality refers to limiting information access and disclosure to only authorized users, as well as preventing access by, or disclosure to, unauthorized ones. The possible values for this metric are listed in Table 4. Increased confidentiality impact increases the vulnerability score.

3.2.5.1 None (N)

There is no impact to the confidentiality of the system.

3.2.5.2 Partial (P)

There is considerable informational disclosure. Access to some system files is possible, but the attacker does not have control over what is obtained, or the scope of the loss is constrained. An example is a vulnerability that divulges only certain tables in a database.

3.2.5.3 Complete (C)

There is total information disclosure, resulting in all system files being revealed. The attacker is able to read all of the system's data (memory, files, etc.)

3.2.6. Integrity Impact (I)

3.2.6.1 None (N)

There is no impact to the integrity of the system.

3.2.6.2 Partial (P)

Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited. For example, system or application files may be overwritten or modified, but either the attacker has no control over which files are affected or the attacker can modify files within only a limited context or scope.

3.2.6.3 Complete (C)

There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised. The attacker is able to modify any files on the target system.

3.2.7. Availability Impact (A)

3.2.7.1 None (N)

There is no impact to the availability of the system.

3.2.7.2 Partial (P)

There is reduced performance or interruptions in resource availability. An example is a network-based flood attack that permits a limited number of successful connections to an Internet service.

3.2.7.3 Complete (C)

There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.

3.2.8. Total severity ranking

CVSSv2 provides severity rankings of "Low," "Medium," and "High" in addition to the numeric scores. These qualitative rankings are mapped from the numeric scores.

Severity CVSS score	**Low** 0.0 – 3.9	**Medium** 4.0 – 6.9	**High** 7.0 – 10.0

For more information, please refer to: http://www.first.org/cvss/cvss-guide.html