Internet of Things Top Ten

Agenda

- Introduction
- Misconception
- Considerations
- The OWASP Internet of Things Top 10 Project
- The Top 10 Walkthrough

26 Billion by 2020

- 30 fold increase from 2009 in Internet of Things install base
- Revenue exceeding $300 billion in 2020
- $1.9 trillion in global economic impact

*Gartner Internet of Things Report 2013

Misconception | It’s all about the device

- It’s not just about the device, or the network, or the clients
- There are MANY surface areas involved
- Each of these need to be evaluated

Considerations | A holistic approach is required

- All elements need to be considered
  - The Internet of Things Device
  - The Cloud
  - The Mobile Application
  - The Network Interfaces
  - The Software
  - Use of Encryption
  - Use of Authentication
  - Physical Security
  - USB ports
- Enter the OWASP Internet of Things Top Ten Project

Internet of Things Top Ten Project | A complete IoT Review

- Review all aspects of Internet of Things
- Top Ten Categories
- Covers the entire device
- Without comprehensive coverage like this it would be like getting your physical but only checking one arm
- We must cover all surface area to get a good assessment of overall security

OWASP
Open Web Application Security Project

I1 | Insecure Web Interface

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability EASY	Prevalence COMMON	Detectability EASY	Impact SEVERE	Application / Business Specific
Consider anyone who has access to the web interface including internal and external users.	Attacker uses weak credentials, captures plain-text credentials or enumerates accounts to access the web interface. Attack could come from external or internal users.	An insecure web interface can be present when issues such as account enumeration, lack of account lockout or weak credenitals are present. Insecure web interfaces are prevalent as the intent is to have these interfaces exposed only on internal networks, however threats from the internal users can be just as significant as threats from external users. Issues with the web interface are easy to discover when examining the interface manually along with automated testing tools to identify other issues such as cross-site scripting.		Insecure web interfaces can result in data loss or corruption, lack of accountability, or denial of access and can lead to complete device takeover.	Consider the business impact of poorly secured web interfaces that could lead to compromised devices along with compromised customers. Could your customers be harmed? Could your brand be harmed?

I1 | Insecure Web Interface | Testing

Is My Web Interface Secure?

Checking for an Insecure Web Interface includes:

- Determining if the default username and password can be changed during initial product setup
- Determining if a specific user account is locked out after 3 - 5 failed login attempts
- Determining if valid accounts can be identified using password recovery mechanisms or new user pages
- Reviewing the interface for issues such as cross-site scripting, cross-site request forgery and sql injection.

Example Attack Scenarios

Scenario #1: The web interface presents "Forgot Password" functionality which upon entering an invalid account informs the attacker that the account does not exist. Once valid accounts are identified, password guessing can begin for an indefinite amount of time if no account lockout controls exist.

```
Account john@doe.com does not exist.
```

Scenario #2: Web interface is susceptible to cross-site scripting.

```
http://xyz.com/index.php?user=<script>alert(123)
</script> ... Response from browser is an alert
popup.
```

In the cases above, the attacker is able to easily determine if an account is valid or not and is also able to determine that the site is susceptible to cross-site scripting (XSS).

- Account Enumeration
- Weak Default Credentials
- Credentials Exposed in Network Traffic
- Cross-site Scripting (XSS)
- SQL-Injection
- Session Management
- Account Lockout

OWASP
Open Web Application Security Project

I1 | Insecure Web Interface | Make It Secure

How Do I Make My Web Interface Secure?

A secure web interface requires:

1. Default passwords and ideally default usernames to be changed during initial setup
2. Ensuring password recovery mechanisms are robust and do not supply an attacker with information indicating a valid account
3. Ensuring web interface is not susceptible to XSS, SQLi or CSRF
4. Ensuring credentials are not exposed in internal or external network traffic
5. Ensuring weak passwords are not allowed
6. Ensuring account lockout after 3 -5 failed login attempts

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer

I2 | Insufficient Authentication/Authorization

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability AVERAGE	Prevalence COMMON	Detectability EASY	Impact SEVERE	Application / Business Specific
Consider anyone who has access to the web interface, mobile interface or cloud interface including internal and external users.	Attacker uses weak passwords, insecure password recovery mechanisms, poorly protected credentials or lack of granular access control to access a particular interface. Attack could come from external or internal users.	Authentication may not be sufficient when weak passwords are used or are poorly protected. Insufficient authentication/authorization is prevalent as it is assumed that interfaces will only be exposed to users on internal networks and not to external users on other networks. Deficiencies are often found to be present across all interfaces. Many Issues with authentication/authorization are easy to discover when examining the interface manually and can also be discovered via automated testing.		Insufficient authentication/authorization can result in data loss or corruption, lack of accountability, or denial of access and can lead to complete compromise of the device and/or user accounts.	Consider the business impact of compromised user accounts and possibly devices. All data could be stolen, modified, or deleted. Could your customers be harmed?

I2 | Insufficient Authentication/Authorization | Testing

Is My Authentication/Authorization Sufficient?

Checking for Insufficient Authentication includes:

- Attempting to use simple passwords such as "1234" is a fast and easy way to determine if the password policy is sufficient across all interfaces
- Reviewing network traffic to determine if credentials are being transmitted in clear text
- Reviewing requirements around password controls such as password complexity, password history check, password expiration and forced password reset for new users
- Reviewing whether re-authentication is required for sensitive features

Checking for Insufficient Authorization includes:

- Reviewing the various interfaces to determine whether the interfaces allow for separation of roles. For example, all features will be accessible to administrators, but users will have a more limited set of features available.
- Reviewing access controls and testing for privilege escalation

Example Attack Scenarios

Scenario #1: The interface only requires simple passwords.

```
Username = Bob; Password = 1234
```

Scenario #2: Username and password are poorly protected when transmitted over the network.

```
Authorization: Basic YWRtaW46MTIzNA==
```

In the cases above, the attacker is able to either easily guess the password or is able to capture the credentials as they cross the network and decode it since the credentials are only protected using Base64 Encoding.

- Lack of Password Complexity
- Poorly Protected Credentials
- Lack of Two Factor Authentication
- Insecure Password Recovery
- Privilege Escalation
- Lack of Role Based Access Control

OWASP
Open Web Application Security Project

I2 | Insufficient Authentication/Authorization | Make It Secure

How Do I Make My Authentication/Authorization Better?

Sufficient authentication/authorization requires:

1. Ensuring that the strong passwords are required
2. Ensuring granular access control is in place when necessary
3. Ensuring credentials are properly protected
4. Implement two factor authentication where possible
5. Ensuring that password recovery mechanisms are secure
6. Ensuring re-authentication is required for sensitive features
7. Ensuring options are available for configuring password controls

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer

OWASP
Open Web Application Security Project

I3 | Insecure Network Services

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability AVERAGE	Prevalence UNCOMMON	Detectability AVERAGE	Impact MODERATE	Application / Business Specific
Consider anyone who has access to the device via a network connection, including external and internal users.	Attacker uses vulnerable network services to attack the device itself or bounce attacks off the device. Attack could come from external or internal users.	Insecure network services may be susceptible to buffer overflow attacks or attacks that create a denial of service condition leaving the device inaccessible to the user. Denial of service attacks against other users may also be facilitated when insecure network services are available. Insecure network services can often be detected by automated tools such as port scanners and fuzzers.		Insecure network services can result in data loss or corruption, denial of service or facilitation of attacks on other devices.	Consider the business impact of devices which have been rendered useless from a denial of service attack or the device is used to facilitate attacks against other devices and networks. Could your customers or other users be harmed?

I3 | Insecure Network Services | Testing

Are My Network Services Secure?

Checking for Insecure Network Services includes:

- Determining if insecure network services exist by reviewing your device for open ports using a port scanner
- As open ports are identified, each can be tested using any number of automated tools that look for DoS vulnerabilities, vulnerabilities related to UDP services and vulnerabilities related to buffer overflow and fuzzing attacks
- Reviewing network ports to ensure they are absolutely necessary and if there are any ports being exposed to the internet using UPnP.

Example Attack Scenarios

Scenario #1: Fuzzing attack causes network service and device to crash.

```
GET %s%s%s%s%s%s%s%s%s%s%s%s%s%s%s HTTP/1.0
```

Scenario #2: Ports open to the internet possibly without the user's knowledge via UPnP.

```
Port 80 and 443 exposed to the internet via a
home router.
```

In the cases above, the attacker is able to disable the device completely with an HTTP GET or access the device via the internet over port 80 and/or port 443.

- Vulnerable Services
- Buffer Overflow
- Open Ports via UPnP
- Exploitable UDP Services
- Denial-of-Service
- DoS via Network Device Fuzzing

OWASP
Open Web Application Security Project

I3 | Insecure Network Services | Make It Secure

How Do I Secure My Network Services?

Securing network services requires:

1. Ensuring only necessary ports are exposed and available.
2. Ensuring services are not vulnerable to buffer overflow and fuzzing attacks.
3. Ensuring services are not vulnerable to DoS attacks which can affect the device itself or other devices and/or users on the local network or other networks.
4. Ensuring network ports or services are not exposed to the internet via UPnP for example

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer

I4 | Lack of Transport Encryption

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability AVERAGE	Prevalence COMMON	Detectability EASY	Impact SEVERE	Application / Business Specific
Consider anyone who has access to the network the device is connected to, including external and internal users.	Attacker uses the lack of transport encryption to view data being passed over the network. Attack could come from external or internal users.	Lack of transport encryption allows data to be viewed as it travels over local networks or the internet. Lack of transport encryption is prevalent on local networks as it is easy to assume that local network traffic will not be widely visible, however in the case of a local wireless network, misconfiguration of that wireless network can make traffic visible to anyone within range of that wireless network. Many Issues with transport encryption are easy to discover simply by viewing network traffic and searching for readable data. Automated tools can also look for proper implementation of common transport encryption such as SSL and TLS.		Lack of transport encryption can result in data loss and depending on the data exposed, could lead to complete compromise of the device or user accounts.	Consider the business impact of exposed data as it travels across various networks. Data could be stolen or modified. Could your users be harmed by having their data exposed?

I4 | Lack of Transport Encryption | Testing

Do I Use Transport Encryption?

Checking for Lack of Transport Encryption includes:

- Reviewing network traffic of the device, its mobile application and any cloud connections to determine if any information is passed in clear text
- Reviewing the use of SSL or TLS to ensure it is up to date and properly implemented
- Reviewing the use of any encryption protocols to ensure they are recommended and accepted

Example Attack Scenarios

Scenario #1: The cloud interface uses only HTTP.

```
http://www.xyzcloudsite.com
```

Scenario #2: Username and password are transmitted in the clear over the network.

```
http://www.xyzcloud.com/login.php?userid=3&
password=1234
```

In the cases above, the attacker has the ability to view sensitive data in the clear due to lack of transport encryption.

- Unencrypted Services via the Internet
- Unencrypted Services via the Local Network
- Poorly Implemented SSL/TLS
- Misconfigured SSL/TLS

OWASP
Open Web Application Security Project

I4 | Lack of Transport Encryption | Make It Secure

How Do I Use Transport Encryption?

Sufficient transport encryption requires:

1. Ensuring data is encrypted using protocols such as SSL and TLS while transiting networks.
2. Ensuring other industry standard encryption techniques are utilized to protect data during transport if SSL or TLS are not available.
3. Ensuring only accepted encryption standards are used and avoid using proprietary encryption protocols

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer

I5 | Privacy Concerns

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	**Exploitability AVERAGE**	**Prevalence COMMON**	**Detectability EASY**	**Impact SEVERE**	**Application / Business Specific**
Consider anyone who has access to the device itself, the network the device is connected to, the mobile application and the cloud connection including external and internal users.	Attacker uses multiple vectors such as insufficient authentication, lack of transport encryption or insecure network services to view personal data which is not being properly protected or is being collected unnecessarily. Attack could come from external or internal users.	Privacy concerns generated by the collection of personal data in addition to the lack of proper protection of that data is prevalent. Privacy concerns are easy to discover by simply reviewing the data that is being collected as the user sets up and activates the device. Automated tools can also look for specific patterns of data that may indicate collection of personal data or other sensitive data.		Collection of personal data along with a lack of protection of that data can lead to compromise of a user's personal data.	Consider the business impact of personal data that is collected unnecessarily or isn't protected properly. Data could be stolen. Could your customers be harmed by having this personal data exposed?

I5 | Privacy Concerns | Testing

Does My Device Present Privacy Concerns?

Checking for Privacy Concerns includes:

- Identifying all data types that are being collected by the device, its mobile app and any cloud interfaces
- The device and it's various components should only collect what is necessary to perform its function
- Personally identifiable information can be exposed when not properly encrypted while at rest on storage mediums and during transit over networks
- Reviewing who has access to personal information that is collected

Example Attack Scenarios

Scenario #1: Collection of personal data.

```
Date of birth, home address, phone number, etc.
```

Scenario #2: Collection of financial and/or health information.

```
Credit card data and bank account information.
```

In the cases above, exposure of any of the data examples could lead to identity theft or compromise of accounts.

- Collection of Unnecessary Personal Information

OWASP
Open Web Application Security Project

I5 | Privacy Concerns | Make It Secure

How Do I Prevent Privacy Concerns?

Minimizing privacy concerns requires:

1. Ensuring only data critical to the functionality of the device is collected
2. Ensuring any data collected is properly protected with encryption
3. Ensuring the device and all of its components properly protect personal information
4. Ensuring only authorized individuals have access to collected personal information

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer

I6 | Insecure Cloud Interface

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability AVERAGE	Prevalence COMMON	Detectability EASY	Impact SEVERE	Application / Business Specific
Consider anyone who has access to the internet.	Attacker uses multiple vectors such as insufficient authentication, lack of transport encryption and account enumeration to access data or controls via the cloud website. Attack will most likely come from the internet.	An insecure cloud interface is present when easy to guess credentials are used or account enumeration is possible. Insecure cloud interfaces are easy to discover by simply reviewing the connection to the cloud interface and identifying if SSL is in use or by using the password reset mechanism to identify valid accounts which can lead to account enumeration.		An insecure cloud interface could lead to compromise of user data and control over the device.	Consider the business impact of an insecure cloud interface. Data could be stolen or modified and control over devices assumed. Could your customers be harmed? Could your brand be harmed?

I6 | Insecure Cloud Interface | Testing

Is My Cloud Interface Secure?

Checking for a Insecure Cloud Interface includes:

- Determining if the default username and password can be changed during initial product setup
- Determining if a specific user account is locked out after 3 - 5 failed login attempts
- Determining if valid accounts can be identified using password recovery mechanisms or new user pages
- Reviewing the interface for issues such as cross-site scripting, cross-site request forgery and sql injection.
- Reviewing all cloud interfaces for vulnerabilities (API interfaces and cloud-based web interfaces)

Example Attack Scenarios

Scenario #1: Password reset indicates whether account is valid.

```
Password Reset "That account does not exist."
```

Scenario #2: Username and password are poorly protected when transmitted over the network.

```
Authorization: Basic S2ZjSDFzYkF4ZzoxMjM0NTY3
```

In the cases above, the attacker is able to either determine a valid user account or is able to capture the credentials as they cross the network and decode them since the credentials are only protected using Base64 Encoding.

- Account Enumeration
- No Account Lockout
- Credentials Exposed in Network Traffic

OWASP
Open Web Application Security Project

I6 | Insecure Cloud Interface | Make It Secure

How Do I Secure My Cloud Interface?

A secure cloud interface requires:

1. Default passwords and ideally default usernames to be changed during initial setup
2. Ensuring user accounts can not be enumerated using functionality such as password reset mechanisms
3. Ensuring account lockout after 3- 5 failed login attempts
4. Ensuring the cloud-based web interface is not susceptible to XSS, SQLi or CSRF
5. Ensuring credentials are not exposed over the internet
6. Implement two factor authentication if possible

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer

OWASP
Open Web Application Security Project

I7 | Insecure Mobile Interface

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability AVERAGE	Prevalence COMMON	Detectability EASY	Impact SEVERE	Application / Business Specific
Consider anyone who has access to the mobile application.	Attacker uses multiple vectors such as insufficient authentication, lack of transport encryption and account enumeration to access data or controls via the mobile interface.	An insecure mobile interface is present when easy to guess credentials are used or account enumeration is possible. Insecure mobile interfaces are easy to discover by simply reviewing the connection to the wireless networks and identifying if SSL is in use or by using the password reset mechanism to identify valid accounts which can lead to account enumeration.		An insecure mobile interface could lead to compromise of user data and control over the device.	Consider the business impact of an insecure mobile interface. Data could be stolen or modified and control over devices assumed. Could your customers be harmed? Could your brand be harmed?

OWASP
Open Web Application Security Project

I7 | Insecure Mobile Interface | Testing

Is My Mobile Interface Secure?

Checking for an Insecure Mobile Interface includes:

- Determining if the default username and password can be changed during initial product setup
- Determining if a specific user account is locked out after 3 - 5 failed login attempts
- Determining if valid accounts can be identified using password recovery mechanisms or new user pages
- Reviewing whether credentials are exposed while connected to wireless networks
- Reviewing whether two factor authentication options are available

Example Attack Scenarios

Scenario #1: Password reset indicates whether account exist or not.

```
Password Reset "That account does not exist."
```

Scenario #2: Username and password are poorly protected when transmitted over the network.

```
Authorization: Basic S2ZjSDFzYkF4ZzoxMjM0NTY3
```

In the cases above, the attacker is able to either determine a valid user account or is able to capture the credentials as they cross the network and decode them since the credentials are only protected using Base64 Encoding.

- Account Enumeration
- No Account Lockout
- Credentials Exposed in Network Traffic

I7 | Insecure Mobile Interface | Make It Secure

How Do I Secure My Mobile Interface?

A secure mobile interface requires:

1. Default passwords and ideally default usernames to be changed during initial setup
2. Ensuring user accounts can not be enumerated using functionality such as password reset mechanisms
3. Ensuring account lockout after an 3 - 5 failed login attempts
4. Ensuring credentials are not exposed while connected to wireless networks
5. Implementing two factor authentication if possible

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer

I8 | Insufficient Security Configurability

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability AVERAGE	Prevalence COMMON	Detectability EASY	Impact MODERATE	Application / Business Specific
Consider anyone who has access to the device.	Attacker uses the lack of granular permissions to access data or controls on the device. The attacker could also us the lack of encryption options and lack of password options to perform other attacks which lead to compromise of the device and/or data. Attack could potentially come from any user of the device whether intentional or accidental.	Insufficient security configurability is present when users of the device have limited or no ability to alter its security controls. Insufficient security configurability is apparent when the web interface of the device has no options for creating granular user permissions or for example, forcing the use of strong passwords. Manual review of the web interface and its available options will reveal these deficiencies.		Insufficient security configurability could lead to compromise of the device whether intentional or accidental and/or data loss.	Consider the business impact if data can be stolen or modified and control over the device assumed. Could your customers be harmed?

I8 | Insufficient Security Configurability | Testing

Is My Security Configurability Sufficient?

Checking for Insufficient Security Configurability includes:

- Reviewing the administrative interface of the device for options to strengthen security such as forcing the creation of strong passwords
- Reviewing the administrative interface for the ability to separate admin users from normal users
- Reviewing the administrative interface for encryption options
- Reviewing the administrative interface for options to enable secure logging of various security events
- Reviewing the administrative interface for options to enable alerts and notifications to the end user for security events

Example Attack Scenarios

Scenario #1: No ability to enforce strong password policies.

```
Admins and users are allowed to create passwords
for their accounts.
```

Scenario #2: No ability to enable encryption of data at rest.

```
Password or other sensitive data stored on the
device may not be encrypted.
```

In the cases above, the attacker is able to use the lack of these controls to get access to user accounts with weak passwords or access data at rest which has protection.

- Lack of Granular Permission Model
- Lack of Password Security Options
- No Security Monitoring
- No Security Logging

OWASP
Open Web Application Security Project

I8 | Insufficient Security Configurability | Make It Secure

How Do I Improve My Security Configurability?

Sufficient security configurability requires:

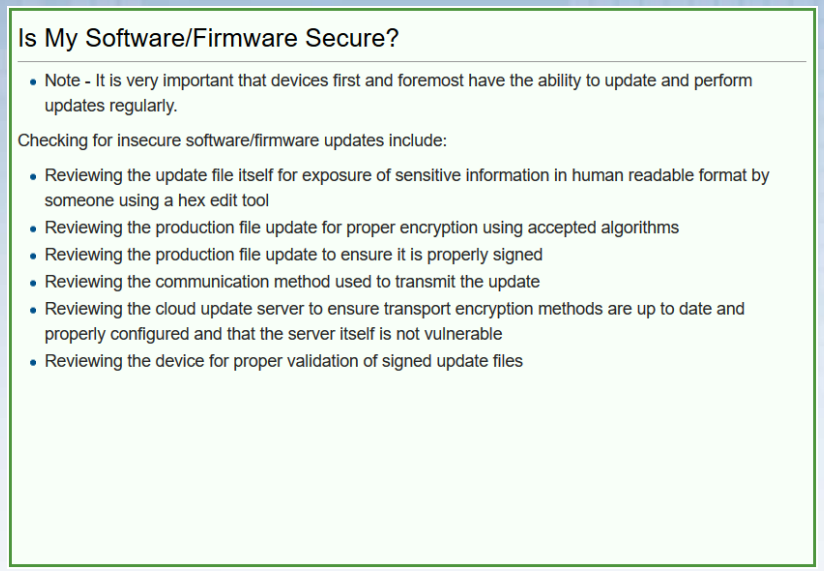
1. Ensuring the ability to separate normal users from administrative users
2. Ensuring the ability to encrypt data at rest or in transit
3. Ensuring the ability to force strong password policies
4. Ensuring the ability to enable logging of security events
5. Ensuring the ability to notify end users of security events

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer

I9 | Insecure Software/Firmware

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability DIFFICULT	Prevalence COMMON	Detectability EASY	Impact SEVERE	Application / Business Specific
Consider anyone who has access to the device and/or the network the device resides on. Also consider anyone who could gain access to the update server.	Attacker uses multiple vectors such as capturing update files via unencrypted connections, the update file itself is not encrypted or they are able to perform their own malicious update via DNS hijacking. Depending on method of update and device configuration, attack could come from the local network or the internet.	The lack of ability for a device to be updated presents a security weakness on its own. Devices should have the ability to be updated when vulnerabilities are discovered and software/firmware updates can be insecure when the updated files themselves and the network connection they are delivered on are not protected. Software/Firmware can also be insecure if they contain hardcoded sensitive data such as credentials. Security issues with software/firmware are relatively easy to discover by simply inspecting the network traffic during the update to check for encryption or using a hex editor to inspect the update file itself for interesting information.		Insecure software/firmware could lead to compromise of user data, control over the device and attacks against other devices.	Consider the business impact if data can be stolen or modified and devices taken control of for the purpose of attacking other devices. Could your customers be harmed? Could other users be harmed?

I9 | Insecure Software/Firmware | Testing

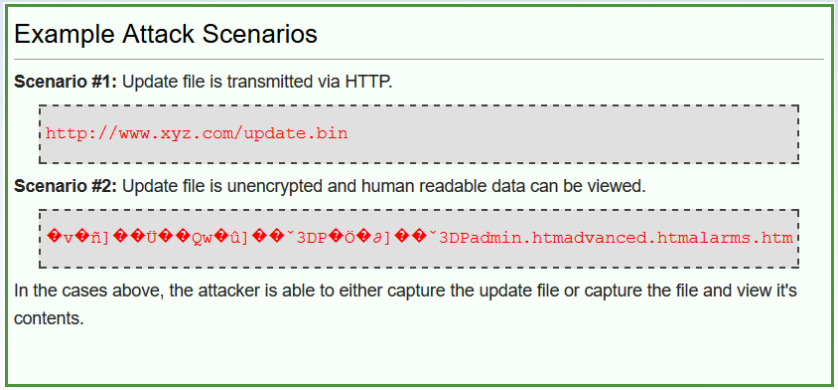


Is My Software/Firmware Secure?

- Note - It is very important that devices first and foremost have the ability to update and perform updates regularly.

Checking for insecure software/firmware updates include:

- Reviewing the update file itself for exposure of sensitive information in human readable format by someone using a hex edit tool
- Reviewing the production file update for proper encryption using accepted algorithms
- Reviewing the production file update to ensure it is properly signed
- Reviewing the communication method used to transmit the update
- Reviewing the cloud update server to ensure transport encryption methods are up to date and properly configured and that the server itself is not vulnerable
- Reviewing the device for proper validation of signed update files



Example Attack Scenarios

Scenario #1: Update file is transmitted via HTTP.

```
http://www.xyz.com/update.bin
```

Scenario #2: Update file is unencrypted and human readable data can be viewed.

```
◆v◆ñ]◆◆Ü◆◆Qw◆û]◆◆ˇ3DP◆Ö◆∂]◆◆ˇ3DPadmin.htmadvanced.htmalarms.htm
```

In the cases above, the attacker is able to either capture the update file or capture the file and view it's contents.

- Encryption Not Used to Fetch Updates
- Update File not Encrypted
- Update Not Verified before Upload
- Firmware Contains Sensitive Information
- No Obvious Update Functionality

I9 | Insecure Software/Firmware | Make It Secure

How Do I Secure My Software/Firmware?

Securing software/firmware require:

1. Ensuring the device has the ability to update (very important)
2. Ensuring the update file is encrypted using accepted encryption methods
3. Ensuring the update file is transmitted via an encrypted connection
4. Ensuring the update file does not contain sensitive data
5. Ensuring the update is signed and verified before allowing the update to be uploaded and applied
6. Ensuring the update server is secure

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer

I10 | Poor Physical Security

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability AVERAGE	Prevalence COMMON	Detectability AVERAGE	Impact SEVERE	Application / Business Specific
Consider anyone who has physical access to the device.	Attacker uses vectors such as USB ports, SD cards or other storage means to access the Operating System and potentially any data stored on the device.	Physical security weaknesses are present when an attacker can disassemble a device to easily access the storage medium and any data stored on that medium. Weaknesses are also present when USB ports or other external ports can be used to access the device using features intended for configuration or maintenance.		Insufficient physical security could lead to compromise of the device itself and any data stored on that device.	Data could be stolen or modified and the device taken control of for purposes other than what was originally intended. Could your customers be harmed? Could your brand be harmed?

I10 | Poor Physical Security | Testing

Is My Physical Security Sufficient?

Checking for Poor Physical Security includes:

- Reviewing how easily a device can be disassembled and data storage mediums accessed or removed
- Reviewing the use of external ports such as USB to determine if data can be accessed on the device without disassembling the device.
- Reviewing the number of physical external ports to determine if all are required for proper device function
- Reviewing the administrative interface to determine if external ports such as USB can be deactivated
- Reviewing the administrative interface to determine if administrative capabilities can be limited to local access only

Example Attack Scenarios

Scenario #1: The device can be easily disassembled and storage medium is an unencrypted SD card.

```
SD card can be removed and inserted into a card
reader to be modified or copied.
```

Scenario #2: USB ports are present on the device.

```
Custom software could be written to take advantage
of features such as updating via the USB port to
modify the original device software.
```

In both cases, an attacker is able to access the original device software and make modifications or simply copy specific target data.

- Access to Software via USB Ports
- Removal of Storage Media

OWASP
Open Web Application Security Project

I10 | Poor Physical Security | Make It Secure

How Do I Physically Secure My Device?

Adequate physical security requires:

1. Ensuring data storage medium can not be easily removed.
2. Ensuring stored data is encrypted at rest.
3. Ensuring USB ports or other external ports can not be used to maliciously access the device.
4. Ensuring device can not be easily disassembled.
5. Ensuring only required external ports such as USB are required for the product to funtion
6. Ensuring the product has the ability to limit administrative capabilities

Please review the following tabs for more detail based on whether you are a Manufacturer, Developer or Consumer