The B Time-to-compromise

by Andrej Zieger
HAW Hamburg / DFN CERT
The B

Time-to-compromise

by Andrej Zieger
HAW Hamburg / DFN CERT
Security by Design...

Guides Best practices and more

Smart Mass

It's more of them than of us

And it is still growing!

A lot effort to get IoT secure!

Security and Resilience of Smart Home Environments
Good practices and recommendations

Vulnerabilities Everywhere!

Research public available data
5 Categories
221 Devices considered
119 Vulnerabilities known

Top Vulnerabilities

<table>
<thead>
<tr>
<th>Count</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>no authentication</td>
</tr>
<tr>
<td>24</td>
<td>DoS</td>
</tr>
<tr>
<td>16</td>
<td>unprotected confidential data</td>
</tr>
<tr>
<td>13</td>
<td>no encryption</td>
</tr>
<tr>
<td>12</td>
<td>unprotected firmware update</td>
</tr>
<tr>
<td>12</td>
<td>hardcoded admin credentials</td>
</tr>
</tbody>
</table>
How we wanna help
Basic Metric
Time-to-compromise by McQueen

3 stochastic processes
1) exploit ready to go
2) try known vulnerabilities
3) search / await zero day

Vuln. count
Skill level
Skill level

\[ TTC(v, s, k) = t_1 \cdot P_1 + t_2 \cdot (1 - P_1) \cdot (1 - u) + t_3 \cdot u \cdot (1 - P_1) \]
Basic Metric
Time-to-compromise by McQueen

\[ TTC(v, s, k) = t_1 \cdot P_1 + t_2 \cdot (1 - P_1) \cdot (1 - u) + t_3 \cdot u \cdot (1 - P_1) \]
Basic Metric
Time-to-compromise by McQueen

Vuln. count
Skill level

Time to tune exploit
\[ c_1 = 1d \]

\[ TTC(v, s, k) = t_1 \cdot P_1 + t_2 \cdot (1 - P_1) \cdot (1 - u) + t_3 \cdot u \cdot (1 - P_1) \]

Probability Proc 1 failed
\[ P_1 = 1 - e^{-v \cdot \frac{m(s)}{k}} \]

Probability Proc 2 failed
\[ u = (1 - f(s))^v \]

Time to try known vulnerabilities
\[ t_2 = c_2 \cdot E(v, s) \]

Time to await zero day
\[ t_3 = \left( \frac{1}{f(s)} - 0.5 \right) \cdot c_3 + c_2 \]

Estimated tries
\[ E(v, s) = f(s) \cdot \left( 1 + \sum_{t=2}^{v-v \cdot f(s)+1} t \cdot \prod_{i=2}^{t} \left( \frac{v \cdot (1 - f(s)) - i + 2}{v - i + 1} \right) \right) \]
What we will improve

Fix it

Continuous skill

Performance

Skill distribution

Compromise type

Exploit complexity

C
I
A
\[ S = \{\text{novice, beginner, intermediate, expert}\} \]

\[ TTC : \mathbb{N} \times S \times \mathbb{N} \to \mathbb{R} \]

\[ m(s) = \text{lookup table} \]

\[ f(s) = \text{lookup table} \]

\[ TTC : \mathbb{N} \times [0, 1] \times \mathbb{N} \to \mathbb{R} \]

\[ f(s) = 0.145 \cdot 2.6^{2s+0.07} - 0.1 \]

\[ m(s) = 83 \cdot 3.5^{4s}/2.7 - 82 \]
\[ E(v, s) = f(s) \cdot \left( 1 + \sum_{t=2}^{v-v \cdot f(s)+1} \left[ t \cdot \prod_{i=2}^{t} \left( \frac{v \cdot (1 - f(s)) - i + 2}{v - i + 1} \right) \right] \right) \]
$E(v, s) = f(s) \cdot \left(1 + \sum_{t=2}^{v-v \cdot f(s)+1} \left[ t \cdot \prod_{i=2}^{t} \left( \frac{v \cdot (1 - f(s)) - i + 2}{v - i + 1} \right) \right] \right)$

$\xi(a, v) = \frac{a}{v} \cdot \left(1 + \sum_{t=2}^{\lfloor v \cdot (1 - \frac{a}{v}) \rfloor + 1} \left[ t \cdot \prod_{i=2}^{t} \left( \frac{v \cdot (1 - \frac{a}{v}) - i + 2}{v - i + 1} \right) \right] \right)$

$E(s, v) = \xi(\lfloor f(s) \cdot v \rfloor, v) \cdot (\lfloor f(s) \cdot v \rfloor - f(s) \cdot v)$

$+ \xi(\lfloor f(s) \cdot v \rfloor, v) \cdot (1 - \lfloor f(s) \cdot v \rfloor + f(s) \cdot v)$
Performance

\[
\xi(a, v) = \frac{a}{v} \cdot \left( 1 + \sum_{t=2}^{\left\lfloor v \cdot (1 - \frac{a}{v}) \right\rfloor + 1} \left[ t \cdot \prod_{i=2}^{t} \left( \frac{v \cdot (1 - \frac{a}{v}) - i + 2}{v - i + 1} \right) \right] \right)
\]

\[
\xi(a, v) = \frac{a}{v} + \frac{a \cdot (v - a)!}{v!} \cdot \sum_{t=2}^{\left\lfloor v \cdot (1 - \frac{a}{v}) \right\rfloor + 1} \left[ t \cdot \frac{(v - t + 1)!}{(v - a - t + 1)! \cdot (v - t + 1)} \right]
\]
Performance Evaluation

TTC0: naive
TTC1: + continuous & fix
TTC2: + using faculty
TTC3: + independent (math complete)
TTC4: + precalculate faculty i < 100

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Min</th>
<th>Max</th>
<th>Avg</th>
<th>TTC/sec</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTC0</td>
<td>13.23s</td>
<td>13.74s</td>
<td>13.37s</td>
<td>150</td>
<td>×1.0</td>
</tr>
<tr>
<td>TTC1</td>
<td>8.41s</td>
<td>8.46s</td>
<td>8.44s</td>
<td>237</td>
<td>×1.6</td>
</tr>
<tr>
<td>TTC2</td>
<td>1.65s</td>
<td>1.67s</td>
<td>1.66s</td>
<td>1206</td>
<td>×8.1</td>
</tr>
<tr>
<td>TTC3</td>
<td>1.06s</td>
<td>1.12s</td>
<td>1.09s</td>
<td>1841</td>
<td>×12.3</td>
</tr>
<tr>
<td>TTC4</td>
<td>0.77s</td>
<td>0.80s</td>
<td>0.78s</td>
<td>2552</td>
<td>×17.1</td>
</tr>
</tbody>
</table>
**Skill Distribution**

Simple idea: skill is a random variable

\[ \int_0^1 TTC(v, s, k) \cdot d(s) \, ds \]

Challenge: can we find one in the wild

Yes! We can.

\[ d = \text{Beta}_{\alpha, \beta}, \quad \alpha = 1.5 \text{ and } \beta = 2 \]

\[ \beta TTC_{\alpha, \beta}(v, k) = \int_0^1 TTC(v, s, k) \cdot \text{Beta}_{\alpha, \beta}(s) \, ds \]
Compromise Type

\[ \mathcal{V}(\alpha, \tau) = \{ y \in \mathcal{W} | y \text{ affects } \alpha \text{ for compromise type } \tau \} \]

Set of all vulnerabilities

An asset

Compromise Type

a vulnerability

Types:
- time-to-compromise-execution (TTCe)
- time-to-compromise-confidentiality (TTCc)
- time-to-compromise-integrity (TTCi)
- time-to-compromise-availability (TTCa)

Example TTCa:
\[ \mathcal{V} = \{ y \in \mathcal{W} | y \text{ affects } a \text{ and } (\text{DAF-effect is DoS or CVSS-availability-impact is not none or CVSS-integrity-impact is complete and DAF-effect is not XSS}) \} \]
**Exploit Complexity**

\[
\text{exc}(y) = 1 - \left( \text{Ac}(y) \cdot \text{Au}(y) \cdot \frac{\text{Ex}(y) - a_1}{a_2} - a_3 \right) \cdot a_4
\]

- **Exploitability**
- **Access complexity**
- **Authentication**

**Influences:**

- Probability of having exploit ready:
  \[
P_1 = 1 - e^{- \left( \sum_{y \in V} 1 - e^{\text{exc}(y)} \right) \cdot \frac{m(s)}{|W|}}
\]

- Fraction of usable vulnerabilities:
  \[
f(s) = \left( \frac{|\{y \in W | s > \text{exc}(y)\}|}{|W|} + a_5 \right) / a_6
\]
Exploit Complexity

Evaluation

bTTC with exc regarding whole DB
Exploit Complexity Evaluation

Exploit complexity differs per product

---

Graph showing the difference in exploit complexity for different operating systems.

- WindowsServer2012
- AppleOSX
- Windows10
- Linux
- approx. McQueen [5]
Further Work

Further CTO

Networks & complex systems

Asset Values
Thanks / Credit

Co-Authors
Felix Freiling - FAU Erlangen-Nürnberg
Klaus-Peter Kossakowski - HAW Hamburg

Resources
Clock - designed by freepik.com
Smart Mass statistics - Gartner

Critics & Corrections
Lutz Euler  Sagar Gurditta
Jan Kohlrausch  Moritz Duge
Alex Mantel