10
Commonly Overlooked Security Hazards

David Piscitello
Core Competence, Inc.
dave@corecom.com
http://hhi.corecom.com
http://www.corecom.com

What we will discuss…

- The 10 commonly overlooked security hazards
- Simple ways to prevent them from placing your network at risk

What? No SANS Top 20?
No clever config tricks?

- You want a quick fix?
  A kewl pen-testing script?
  You came to the wrong room!
- 2-minute response to “Fixing security holes”
  - Patching holes is a necessary activity
  - Not a sufficient strategy for lifeboats or security…
  - Does not address root causes
- The most important aspects of security are low-tech

10 commonly overlooked security hazards

1. Lax policy definition and enforcement
2. Overly permissive access policies
3. Single lines of defense
4. Default installations of software
5. Default and vulnerable configurations
6. Weak authentication methods
7. Inadequate auditing, logging, analysis
8. Flawed security processes, un-secured workflows
9. Weak security testing and auditing methodologies
10. Weak incident response & business continuity plans

1. Lax policy definition and enforcement

- No clear (documented) understanding of
  - Assets and their value
  - Whether assets are vulnerable and how
  - What risk vulnerabilities pose
- Security implementation is changed first, policy is adjusted later (maybe…)
- No dissemination of policy to stake-holders
- No compliance
- No accountability
- No enforcement

The problems caused when policy is neglected

- You don’t really know what you’re securing and why
- You spend $$$ on security without direction
- Changes to policy go undocumented
  - Risk analysis is neglected
  - Impact of changes impossible to verify
  - Processes affected by change may not be changed
- You have nothing on which to base appropriate use
  - Stakeholders do what they think is OK
  - Default policy is “Ask forgiveness, not permission”
  - Stakeholders cannot be held accountable

The simple fix

- Develop and maintain a security policy
- A security policy says:
  - "Here is what we value, how we intend to protect it, and what we will do if it should be lost, damaged, or attacked."
- Document procedures for
  - Appropriate use and handling of assets
  - What constitutes authorized access
  - Maintaining security as networks and needs change
  - Responding to attacks or incidents

2. Overly permissive Internet access policies

- "More" is NOT better
  - Super-sizing your Internet access is A Bad Idea
- Examples:
  - All users are provided the same level of access
    - ANY internal and Internet services, from ANY location
  - A firewall's default policy is ALLOW ANY outbound
  - File and printer sharing is public/anonymous/ANY

The problems "allow any" access causes

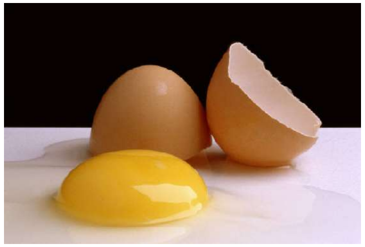
- Unauthorized access
- Disclosure of sensitive information
- Unintended download of malware
  - Virus infections and spyware pest infestations
- Unintended upload of privacy information
  - Back channel communication from infested PCs to spyware and adware servers
- Unanticipated administrative assistance
  - Remote administration and rogue operation by attackers

The simple fix

- Implement stronger authorization
- Grant permission based on strongest authentication possible (even for Internet access)
- Follow the Law of Least Privilege:
  Only grant individuals access to what they
  need to do their jobs

3. Single Line of Defense

- **Internet Firewalls no longer keep outsiders at bay**
  - **Mobile workers, day-extenders, WLANs, and business relationships makes “outsider” hard to identify**
- **Learn from the Maginot Line…**
  - **Beware of an end-run around a long line of forts**
- **Analogy for the history-impaired: Does your security resemble soft-boiled egg?**
  - **Hard on the outside, soft in the middle**

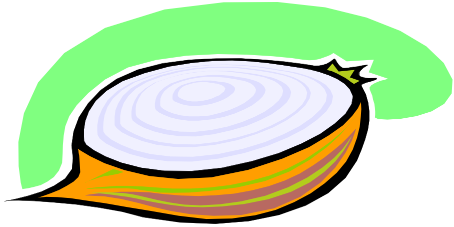


Your Server ?

The problems monolithic defenses cause

- Attackers “end-run” around your defenses
- VPN tunnels become highways for attackers
- Eavesdropping on non-work WLANs
- Non-work endpoints are ripe for malware

The simple fix



- An onion offers a better analogy
  - Defense in depth…
- Apply defenses at all Internet architecture layers
  - Physical, Link, Network, Transport, Application
- Build concentric rings of defense (Edwardian Castles)
  - Anti-malware at gateway, client, and server
  - Firewalls at gateway, client, and server
  - Anti-tampering and HIDS on clients and servers
  - Device- and user-level authentication
  - Admission control for managed and unmanaged systems

4. Default installations of software

- Majority of software installs to ‘plug and play’
  - Anyone can play
  - Any application they choose
  - Even ones you didn’t intend to offer
- Examples:
  - Windows default startup services
    - Messenger, Remote Registry Service, Secondary Logon
  - Grandstream SIP phone
    - tftp is listening
  - Many SOHO firewall, NAT, broadband routers
    - HTTP management (not SSL-protected)

Problems default installs cause

- Services and applications run with default permissions and configurations
  - Leak information
  - Are not audited
  - Accept anonymous connections
  - Provide opportunities to exploit test and example scripts
- These can lead to escalated privilege attacks

The simple fix

- Document the default operating mode of every system you run
- Define what you need in a policy
- Run what you need, turn everything else off!
    - Disable unnecessary services (esp. on clients)
    - Restrict/prohibit services on client PCs
    - Routinely scan systems for listening services
- Only add services when policy is revised

5. Default and vulnerable configurations

- Network devices want to create and join networks
  - Open policies facilitate ‘instant networking’
  - Open to and for all is a poor baseline for securing networks
- Examples:
  - A WLAN AP defaults to open architecture
  - A router or switch runs with SNMP enabled (Get/READ)
  - Windows default account has full (administrator) privileges
  - Web and ftp server banner identify OS and server types and versions

The problems default configs cause

- Bandwidth abuse
- Eavesdropping
- Information gathering
- DOS attacks
- Unauthorized access
- User self-administration often facilitates auto-installation of malicious code

The simple fix

- Do not put devices into production until default configurations have been removed
  - Vulnerability assessment tools scan for defaults
- Block everything initially; allow services defined by policy and no others
- Restrict/prohibit user self-administration
  - Never run as admin unless you are administering

6. Weak authentication methods

- Passwords are simple to derive, especially when you
  - Share them
  - Write them down on Post-Its
  - Save them in your browser
  - Use the same password for e-tailing, e-banking, and your extranets and intranets
  - Enter them in any form that asks with impunity☺
- Two-factor authentication is better, unless you
  - Velcro the token to a monitor, next to the Post-It where you wrote the PIN
  - Write the PIN on the back of the token
- Biometrics are better
  - Until your templated body part is used against your will or under duress

The problems weak authentication causes

- **Misuse of account by unauthorized (but authenticated) individuals**
- **Impersonation and forgery**
- **Unauthorized access to sensitive data**
- **What's the root cause?**

Low-Tech Password Cracker: Chocolate
April 20, 2004
By Enterprise IT Planet Staff

Trade your password for a bar of chocolate? You would probably (and responsibly) decline, but some Londoners took up the offer.

Out of a small sample of 172 office workers that were approached on the street, more than a third (37%) willingly divulged their password when simply asked, according to Infosecurity Europe 2004's organizers. Sadly, a large majority -- a full 71 percent -- forked over the information when bribed with chocolate.

The simple fix (social)

- Authentication is as much a social as a technology problem
- Correct social problems through behavior modification
  - Educate users about social engineering,
  - Teach users proper password maintenance
  - Anti-phishing initiatives and remedial education

The simple fix (technology)

- No authentication method is failsafe
  - "...against an opponent that is willing to physically attack, threaten, or torture you, ALL authentication systems are worthless!" – Marcus Ranum
- Any authentication method can be used effectively
  - Creating sufficient resilience against probable attack is 10% of the solution
  - Compliance is the other 90%

7. Inadequate auditing, logging, analysis

- Auditing is not an in-depth activity
  - Too few audit points in the network
  - Too little information is audited
  - What is audited has more to do with accounting than security
- Audit information is not
  - Aggregated
  - Cross-correlated
  - Analyzed
  - Verified and protected against tampering

The problems poor auditing, logging, and analysis cause

- You can’t easily confirm your implementation conforms to your policy
- You have no idea who’s connected to, and what is running on, your network
- You cannot distinguish normal from abnormal behavior (abuse, attack)
- You cannot relate security events that occur on multiple systems at multiple locations
- You cannot rely on audit data accuracy for incident response or legal action
- You cannot demonstrate you made a “best effort” to comply with regulations

The simple fix

- **Perform auditing at many levels:**
  - **User, operating system: Login attempts, policy violations**
  - **Network protocol: Connection attempts, malformed packets**
  - **Network equipment: Route changes, management logins**
  - **Security systems: Policy violations, intrusion attempts**

OK, I lied, it's not that simple…

- Synchronize time to facilitate cross-correlation of events
- Tamper-proof audit records
  - Otherwise, records are of no value to forensics and may not be suitable as evidence
- Develop a companion analysis process
- Use auditing and analysis proactively
  - Important for IR, but also useful for predictive analysis

8. Flawed security processes, un-secured workflows

- Processes that should be “atomic events” but
  - Require manual implementation and sign-off
  - Rely on single authority at multiple sign-off levels
  - Cannot be (easily) undone or readily reproduced
- Security related processes that
  - Can be eavesdropped or attacked
  - Are not documented and audited
- Examples:
  - Manual or human-driven user registration, archival, removal
  - Remote device administration over un-secured link
  - Any device administration with weak authentication
  - Configuration changes without recovery points

The problems they cause

- Mis-configurations expose assets to attack
- Processes slowed or halted when chain-of-command is unavailable
- Windows of opportunity for disgruntled employees and attackers
- Absence of recovery points makes incident or accident recovery painful and expensive

The simple fix

- Subject all workflows to review
- Automate and audit workflows
- Alert when workflows delayed or interrupted
- Incorporate recovery points into workflows

9. Weak security testing and auditing methodologies

- Poorly documented procedures
- Policy changes not taken into account
- Process is ad hoc
  - Formal methodology forsaken for scans & scripts
  - Compliance guidelines not considered
- Results only used to correct (current) security implementation
- No rigor in execution

The problems they cause

- Testing
  - Is incomplete
  - Is not routinely performed
  - Does not address/mitigate root causes
- Aspects of testing process are not reproducible
- Testing and policy changes are not associated events
  - Auditing is challenging in such situations and like testing, is incomplete and can’t help identify root causes
- Auditing does not meet criteria set by regulators

The simple fix

- Develop a formal methodology
  - Establish relationship between testing and policy management, and audit against policy
- Document each test
  - What is to be tested
  - Expected versus actual results
  - Prioritize remedial activities at implementation level
  - Test frequency and scheduling
- Focus on root cause rather than symptoms
- Input results of analysis to policy management/definition process

10. Weak incident response and business continuity plans

- No documented procedures for
  - Responding to incidents
  - Containing the damage
  - Preserving “state” and evidence
  - Escalating the response
  - Engaging law enforcement
  - Disclosure of the incident to public, shareholders, regulators, and customers
  - Continuing operations in the face of attack
  - Resuming business should operations halt

The problems they cause

- Chicken Little is not a role model for a CSO
- Valuable time is lost
    - Attack may spread
    - Service outage persists
    - Experts may not be “on call” to respond
- Audit data and potential evidence lost
    - Rebooting is not always a good idea
- Law enforcement response is delayed
- Failure to comply with regulations regarding IR
- Disclosure may not be controlled or accurate

The simple fix

- Develop and disseminate IR and business continuity plans
- Report incidents to law enforcement agencies
- Learn how to work with law enforcement
  - Make your willingness to prosecute public
  - Verify that your security event (audit) data will stand up as evidence in court
  - Prosecute attackers - and PLEASE, don’t hire them!
- Know what regulatory obligations you have
- Engage legal and PR
- Consider preparedness (incident response “fire” drills)
  - There’s less value in discovering you were unprepared after the incident than before

Conclusions

- Many factors contribute to your ability to define and maintain a strong security profile
- The most common hazards to security have less to do with technology than policy and process
- Well-documented policies and processes generally eliminate common security hazards