harborlabs
CYBER . SCIENCE

sample report

The following sample report is based on a fictitious client, product, and security experience. It is provided for informational purposes only, as an example of Harbor Labs' analysis, experimentation and reporting services. While each Harbor Labs report will vary in length and content to meet the unique requirements of each client engagement, this document is nonetheless highly representative of the scope and depth of analytic information that can be expected by Harbor Labs' clients.

Security Review of Extreme Arcade Gaming

Prepared by {A Harbor Labs team member}, Ph.D.
{member}@harborlabs.com

Executive Summary

Harbor Labs sent a proposal for a security review, to include design and architecture review and penetration testing, of Extreme Arcade Gaming's MicroCade v1.2 gaming server. The proposal described Harbor Labs' methodology for conducting the review, and proposed a threat model and security policies. Harbor Labs has executed the review of the security policies via technical discussions with Extreme Arcade Gaming's engineers, reviewing relevant documentation, and testing the gaming server and related systems, such as auxiliary peripherals, for violations against the policies. This testing included reverse engineering, network and serial analysis, web vulnerability scanning, and black-, grey-, and white-box fuzzing. Harbor Labs has discovered weaknesses and vulnerabilities unknown to Extreme Arcade Gaming and has successfully compromised known weaknesses. This document provides the results of Harbor Labs' review. It also includes recommendations for mitigating identified security policy violations, and it suggests standards, tools, and approaches for secure design, implementation, and testing.

sample report

harborlabs
CYBER . SCIENCE

Table of Contents

sample report

harborlabs
CYBER . SCIENCE

Results Summary

POLICY LIST			STATUS
1.		Policies Related to MicroCade Physical Access	
	1.1	No global secrets can be lifted from a gaming server even with physical access.	SEVERE (Known)
2.		Policies Related to MicroCade Configuration and Management	
	2.1	The MicroCade will only install authorized games (i.e., signed game images).	ENFORCED
	2.2	The MicroCade will only connect to authorized auxiliary peripherals (e.g., gaming controllers cannot be spoofed).	
3.		Policies Related to Gaming	
	3.1	A competitive game match cannot be stopped, started, or modified remotely.	SEVERE (Known)
4.		Policies Related to Extreme Arcade Gaming Software and Tools	
	4.1	The Game Uploader software can only be accessed by an authorized user.	UNBROKEN
	4.2	The MicroCade will only communicate with an authorized Game Uploader.	UNBROKEN
5.		Policies Related to Logging	
	5.1	Logging cannot be disabled, interrupted, or modified on the MicroCade.	SEVERE

sample report

harborlabs
CYBER . SCIENCE

I. Introduction

The MicroCade is a gaming server that provides networked gaming access locally or over the Internet for competitive gaming. The MicroCade supports wireless network communication to enable ad-hoc gaming and wireless network access in settings where Ethernet is unavailable. Extreme Arcade Gaming provides proprietary methods to prevent cheating, unauthorized game and software use, and unauthorized peripheral use. In particular, the MicroCade uses real-time memory scanning and multi-gaming server report methods to identify cheating.

We have completed our security review of the Extreme Arcade Gaming MicroCade. This review includes a complete analysis of the MicroCade and related systems. Our analysis was based-on a testing approach. In particular, we tested the MicroCade and related systems for violations against the security policies we identified in the document titled, "A Proposal for Evaluating Extreme Arcade Gaming MicroCade Gaming servers." This testing included reverse engineering, network and serial analysis, web vulnerability scanning, and black-, grey-, and white-box fuzzing.

Prior to our testing, Extreme Arcade Gaming was aware of a number of violated policies. For example, Extreme Arcade Gaming knew that the following policies were violated:

1.1. No global secrets can be lifted from a gaming server even with physical access.

5.1. Logging cannot be disabled, interrupted, or modified on the MicroCade.

The above policy violations are possible because the MicroCade runs an unauthenticated telnet server and hardcodes global secrets such as the administrator password, TLS certificate, and RSA private key. We describe these results in Section III, and we give further technical details in Section IV.

We have identified weaknesses and vulnerabilities that were previously unknown to Extreme Arcade Gaming. For example, we found that an unauthorized user with network access to the MicroCade can bypass cheating methods by accessing the unauthenticated telnet server. This violates the following policy:

3.1. A competitive game match cannot be stopped, started, or modified remotely.

While we describe known and unknown (to Extreme Arcade Gaming) MicroCade weaknesses and vulnerabilities, Extreme Arcade Gaming does have a preliminary plan to mitigate the problems we have identified. We find that their plan is sufficient in enforcing the above policy violations. We discuss this plan in Section V.

The remainder of the document is outlined as follows: first, we provide a brief background of the MicroCade and its related systems in Section II. We focus on the MicroCade architecture because it makes clear what the associated attack surfaces are. Next, we describe our methodology for performing the review in Section III. We described this in the aforementioned proposal document but provide it here for completeness. We give our security policy review (i.e., the results of our review) in Section III, and we provide the associated technical details in Section IV. Lastly, we conclude in Section V with an attack tree for the MicroCade, recommendations, and a review of the preliminary mitigation plan.

harborlabs
CYBER . SCIENCE

II. Background

The MicroCade is a gaming server that provides networked gaming access locally or over the Internet for competitive gaming. The gaming server supports the following features:

- Local multiplayer with multiple auxiliary peripheral support [vector]
- Internet multiplayer with automatic match making implemented in the cloud [vector]
- Competitive game mode which employs cheat detection methods [goal]
- Gamer lockout based on cheat detection methods [goal]
- Wired/Wireless network communication [surface]
- RS-232 communication cable for connecting auxiliary peripherals [vector]
- RF wireless communication interface for connecting wireless auxiliary peripherals [vector]

While not all-inclusive, the features described in the MicroCade user manual are important to our analysis as they represent attack vectors, surfaces, and goals (e.g., an attacker wants to overcome cheat detection methods). We next describe the deployment, use, and architecture of the MicroCade. We recovered details not in the manual and other provided materials via our analysis and reverse engineering.

An authorized user deploys a MicroCade in their home by first configuring the gaming server for use. This configuration is achieved by using one of the attached auxiliary peripherals and the startup software. The startup software enables the authorized user to set and configure network access (i.e., adding keys and passphrases), a user profile that is associated to downloaded games, and register the gaming server to receive software updates over the network.

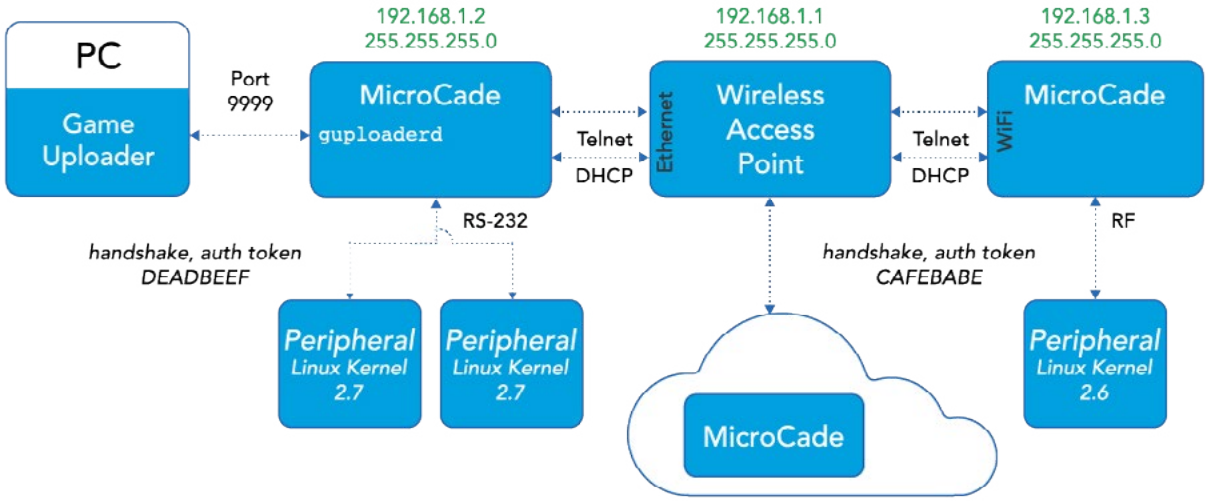

Figure 1: MicroCade network architecture.

As depicted in Figure 1, the MicroCade connects to a network access point via Ethernet or WiFi. The MicroCade also attaches to an auxiliary peripheral via an RS-232 connection or wireless RF signal. The auxiliary peripheral runs an embedded operating system based on the Linux Kernel. The peripheral authenticates to the MicroCade by engaging in a challenge-response protocol. In the protocol, the MicroCade generates a random nonce encrypted with a known public key and the peripheral decrypts that challenge and sends it back. The authorized peripheral's operating system allows the user to customize device functionality by adding pluggable components such as LEDs.

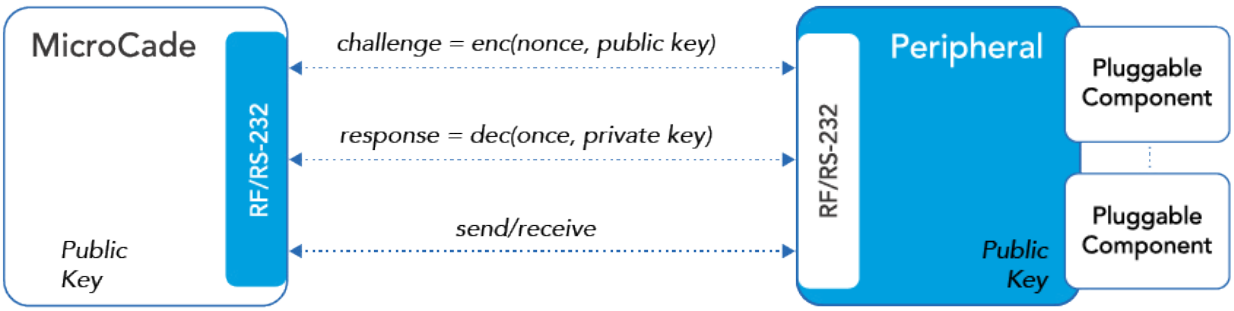

Figure 2: MicroCade and Peripheral challenge-response protocol.

The MicroCade runs a custom operating system that is also based on the Linux Kernel. Unlike the auxiliary peripherals, this operating system features a complete TCP/IP stack. The MicroCade runs a software daemon called the guploaderd. This daemon listens for an authorized connection from a separate computer to install a game. In particular, the user uses the Game Uploader software provided by Extreme Arcade Gaming to load the game on the MicroCade.

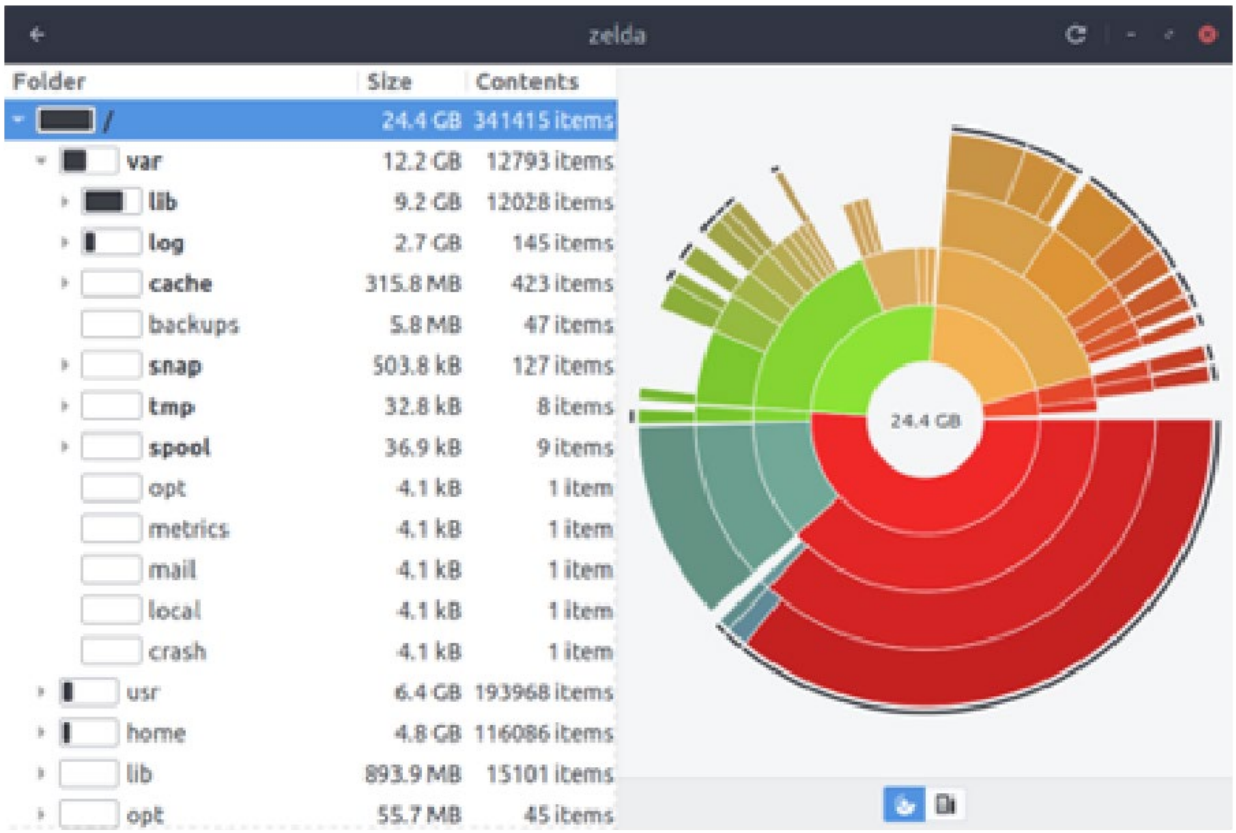

Figure 3: Disk Map of Target System

In contrast with a security review, the Harbor Labs' security evaluation is a more comprehensive exercise. In a security evaluation report, this section would contain additional testing and experimentation information, including the various tools used, their output, and an analysis of how the data supports the security recommendations of the Harbor Lab scientists.

harborlabs
CYBER . SCIENCE

II. Methodology

We analyzed the technical specifications of the MicroCade and identified a set of a security policies as explained in the document titled, "A Proposal for Evaluating Extreme Arcade Gaming MicroCade Gaming servers." Security policies describe expected system invariants (i.e., conditions that are always true) even in the presence of malicious and intentional attacks. For MicroCade, these security policies describe how Extreme Arcade Gaming believes the MicroCade works and how it behaves when attacked. For example, MicroCade should only install authorized games.

We defined multiple threat models that contemplate various scenarios and assumptions about the attacker in the analysis of the MicroCade. The result of these threat models is a set of threat capabilities that we combine or divide for a given security policy. That is, each security policy contains a non-empty set of threat capabilities that describe the attacker.

For each security policy and its associated threat capabilities, we investigated how well the policy was or was not enforced by the MicroCade. We show how we are able to break violated policies by providing technical details. Also, we describe how we might break known violated policies. For these known violated policies, we did not carry out a proof-of-concept.

We note that the ability of the policy to resist one attack does not mean that it is absolutely enforced. Each resisted attack increases confidence in the enforcement of the policy.

Our security policies are based on the review of relevant documentation, execution of the MicroCade and related system, and technical discussions with Extreme Arcade Gaming staff. In the next subsection, we describe our experimental setup, define our threat models, and lastly move to the results of our security policy review.

A. Experimental Setup

Extreme Arcade Gaming provided the required systems to perform our analysis. In particular, we used the following software and hardware:

- Three MicroCade 1.2 gaming servers
- Three auxiliary peripheral (i.e., gaming controller)
- D-Link Cloud Router (DIR-836L)
- Game Uploader application
- A signed Ultimate Fighting game image

- Lab computers (e.g., Mac OS X 10.11)
  - » Wireshark v1.12.6-0 for intercepting network communication
  - » Charles Proxy v3.11.2 for SSL proxy
  - » Access Port v1.37 for sniffing serial connections
  - » RawCap v0.1.5.0 for raw Window's sockets (i.e., loopback)

B. Threat Models

We described our threat models in greater detail in the document titled, "A Proposal for Evaluating Extreme Arcade Gaming MicroCade gaming servers." For completeness, we review our threat models here in terms of our general approach and assumptions.

Our general approach to defining threat models was to identify the capabilities of an attacker. We categorized these capabilities as: network access, physical access, and legitimate or previously legitimate credentials. For each of these categories, there are increasing levels of capability. For example, an attacker that can modify network communication can also insert and eavesdrop on that network communication.

- Network: eavesdropping, insertion, modification
- Physical: physical access, reverse engineering
- Credentials: expired credentials, user credentials, administrator credentials

All of our threat models apply the following standard assumptions.

- Axiomatic Cryptographic Primitives: The attacker cannot break correctly implemented and correctly configured cryptographic primitives (where break is defined within the context of the primitive itself).
- Resources: The attacker may have a significant amount of computational resources on the order of a corporation or large organization (e.g., organized criminal entity).
- Legal: The attacker does not have governmental powers such as subpoena, etc.
- System Knowledge: We assume that the attacker has a full knowledge of system architecture, protocols, and configuration data.

We note that it is important to identify these threat models and make your customers aware of them. For example, if you have an individual policy that assumes that the attacker does not have physical access, the customer must be aware of that as it may change how they deploy the gaming server. We also note that social engineering attacks, which we left out of our current threat model, should not be underestimated. Many hackers have disclosed that social engineering comprises a significant part of their approach.

sample report

III. Security Policy Review

Each policy listed in this section was first described in the document Each of the policies identified in the "detailed proposal" are reviewed here in terms of our ability or inability to violate them. For each policy, we include a tag from the following list:

- ENFORCED – We did not find a violation of the policy.
- WARNING –
  - » It is possible to violate the policy with an expanded threat model (social engineering).
  - » We found a software vulnerability but do not yet know how to exploit it.
  - » Both of the above.
- SEVERE/KNOWN – We confirmed a violation already known to Extreme Arcade Gaming.
- SEVERE – We uncovered a new violation that was not previously known to Extreme Arcade Gaming.

1. Policies Related to MicroCade Physical Access

1.1. No global secrets can be lifted from a gaming server even with physical access.

- SEVERE
- Attack #1:
  - » An attacker can recover the administrator password from the gaming server software image. This password is the same for all MicroCade gaming servers and is hardcoded in the software image.
  - » Threat Class: Administrator credentials and reverse engineering.
- Attack #2:
  - » An attacker can access the unauthenticated telnet server running on the MicroCade. This enables the attacker to recover the TLS private key. This key is the same for all gaming servers and is hardcoded in the software image.
  - » Threat Class: Network insertion and reverse engineering.
- Policy Note #1: We define global secrets as passcodes, private keys, private certificates, long-term encryption keys, and account passwords or tokens.
- Technical Note #1: See Section IV, subsection A4.

2. Policies Related to MicroCade Configuration and Management

2.1. The MicroCade will only install authorized games (i.e., signed software images).

- ENFORCED
- Policy Note #1: We assume that the private key used to sign the game images is not leaked or available outside of Extreme Arcade Gaming's control.
- Technical Note #1: Game images are digitally signed and verified.
- Technical Note #2: The public key is stored on every MicroCade.

2.2. The MicroCade will only connect to authorized auxiliary peripherals (e.g., gaming controllers cannot be spoofed).

- WARNING
- Potential Attack #1:
  - » An attacker with physical access to the MicroCade can plug in an unauthorized controller, remove it, and plug it back in. This bypasses the authorization mechanism, thus allowing an unauthorized peripheral.
  - » Threat Class: Physical access.
- Potential Attack #2:
  - » An attacker can obtain the software image of an authorized auxiliary peripheral and recover the hardcoded private key.
  - » Threat Class: Reverse engineering.
- Policy Note #1: This policy is under review by Extreme Arcade Gaming as they currently describe this as an intended functionality.
- Policy Note #2: The ability to recover the private key is not intended.
- Technical Note #1: See Section IV, subsection A3.

3. Policies Related to Gaming

3.1. A competitive game match cannot be stopped, started, or modified remotely.

- SEVERE

- Attack #1:
  - » An attacker can access the unauthenticated telnet server running on the MicroCade. This enables the attacker to stop the cheat detection processes. The attacker may also modify the cheat detection binaries.
  - » Threat Class: Network insertion and reverse engineering.
  - » Technical Note #1: See Section IV, subsection A2.

4. Policies Related to Extreme Arcade Gaming Software and Tools

4.1. The Game Uploader software can only be accessed by an authorized user.

- ENFORCED
- Policy Note #1: We assume that the software is installed on a Windows or Mac OS X computer that is only accessible by an authorized user.
- Policy Note #2: This is out of the scope of Extreme Arcade Gaming.

4.2. The MicroCade will only communicate with an authorized Game Uploader.

- » Attack #1:
- » An attacker can reverse engineer the Game Uploader software and recover the private key. The attacker can than spoof the Game Uploader.
- » Threat Class: Network insertion and reverse engineering.

- Policy Note #1: Every Game Uploader software has the same private key.
- Policy Note #2: See Section IV, subsection A4.

5. Policies Related to Logging

5.1. Logging cannot be disabled, interrupted, or modified on the MicroCade.

- SEVERE
- Attack #1:
  - » An attacker can access the unauthenticated telnet server running on the MicroCade. This enables the attacker to modify the logging settings.
  - » Threat Class: Network insertion and reverse engineering.
- Policy Note #1: See Section IV, subsection A2.

harborlabs
CYBER . SCIENCE

IV. Technical Analysis Details

We provide technical details for our security policy review. These details describe how we wrote and executed our tests for the MicroCade and related systems. We list technical details in no particular order, but we do group them by system.

A. MicroCade

We give a background for the MicroCade in Section II. In this subsection, we describe our tests for the MicroCade. These tests include a teardown of the device, network analysis of MicroCade communication, serial analysis, and a binary analysis of software provided by Extreme Arcade Gaming.

A1. Physical Access

We had physical access to three MicroCade gaming servers and three gaming controllers. We performed a teardown of each device to learn more about the components that compose each device. We found that the MicroCade is composed of a consumer-grade ATX motherboard and standard components such as:

- 16 GB DDR4 PC4-17000 RAM
- RAID 5 array with 4 1TB 7200RPM hard disks
- 2 GEFORCE GTX 1080 graphic cards
- 4 port PCI RS-232 Serial card

The auxiliary peripheral is shown in the teardown Figure below. It depicts pluggable components such as buttons and a thumb-pad.

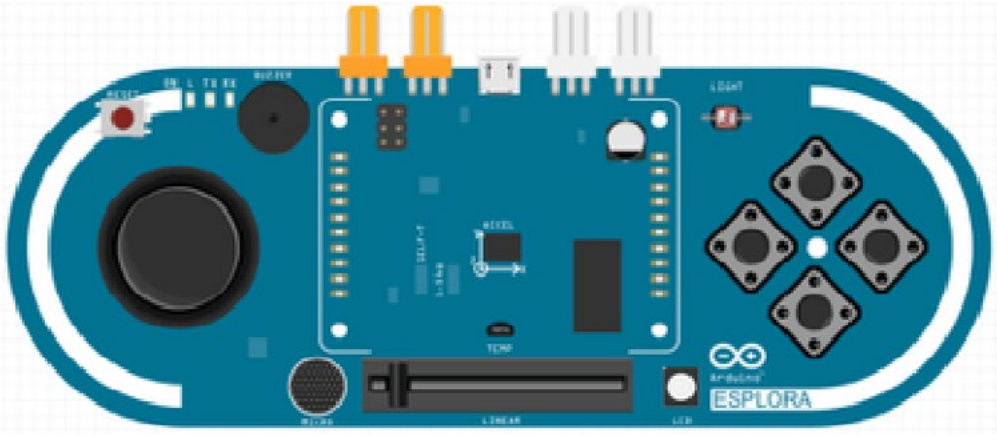

Figure 3: Auxiliary peripheral teardown.

A2. Network Analysis

We performed a network analysis of the MicroCade using Wireshark. We found the telnet server user credentials while passively sniffing the network during a MicroCade remote update. We show the TCP flow in the Figure below.

```
.....  ..!.."..'.....#..%..%.....  ..!.."..".....P.
...."......b........b....       B.
.........................".......'......#..&..&..$..&..&..$..
.....#.....'.....  .9600,9600....#.bam.zing.org:
0.0....'..DISPLAY.bam.zing.org:0.0.......xterm-
color..............!..............".............
OpenBSD/i386 (oof) (ttyp1)

login: .."........"ffaakkee
.
Password:user
.
Last login: Thu Dec  2 21:32:59 on ttyp1 from bam.zing.org
Warning: no Kerberos tickets issued.
OpenBSD 2.6-beta (OOF) #4: Tue Oct 12 20:42:32 CDT 1999

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

$ llss
.
$ llss  --aa
.
.         ..        .cshrc    .login    .mailrc   .profile  .rhosts
$ //ssbbiinn//ppiinngg  wwwwww..yyaahhoooo..ccoomm
.
PING www.yahoo.com (204.71.200.74): 56 data bytes
64 bytes from 204.71.200.74: icmp_seq=0 ttl=239 time=73.569 ms
64 bytes from 204.71.200.74: icmp_seq=1 ttl=239 time=71.099 ms
64 bytes from 204.71.200.74: icmp_seq=2 ttl=239 time=68.728 ms
64 bytes from 204.71.200.74: icmp_seq=3 ttl=239 time=73.122 ms
```

Figure 4: Telnet TCP flow.

We used nmap to discover open ports on the MicroCade. We then implemented a custom network fuzzer that generates garbled TCP packets and sends them to the MicroCade. The source of our network fuzzer is provided in Code Snippet 1.

```
Not shown: 1674 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
79/tcp    open  finger
111/tcp   open  rpcbind
786/tcp   open  concert
5900/tcp  open  vnc
```

Figure 5: Nmap scan of MicroCade.

```
#!/usr/bin/env python


""" NetworkFuzzer.py: Send arbitrary network packets to MicroCade device.
"""


__author__ = "Harbor Labs"

__email__ = "mike@harborlabs.com"


def fuzz(ip, port):

    """ Scapy fuzzing MicroCade gaming server open ports. """

    socket = socket.socket()

    socket.connect((ip, port))

    stream = StreamSocket(socket)


    # This is our fuzzing packet.

    packet = IP(dst=ip)/TCP(dport= port)/fuzz(Raw())


    while(True):

        stream.send(packet)


def main():

    fuzz('192.168.1.2', 23)


if __name__ == "__main__":

    main()
```

Code Snippet 1: MicroCade network fuzzer.

A3. Serial Analysis

We used the Access Port application to sniff the serial connection between the auxiliary peripheral and the MicroCade. We found the bytes DE AD BE EF being sent from the peripheral to the MicroCade when the peripheral was first connected. We concluded that these bytes were handshake, and we were able to replicate the handshake and then send and receive bytes from the MicroCade.

A4. Binary Analysis

We used a combination of the command-line tool binwalk, strings, and IDA Pro to reverse engineer the MicroCade and peripheral software images, and the Game Loader application. We discovered the string superSecretPasssword in the MicroCade binary. This string is the administrator password for the system and is the default for all three MicroCade gaming servers that we received.

V. Conclusion

Extreme Arcade Gaming must decide which policy violations are most severe and in what order these violations will be addressed. We do not have enough domain-specific knowledge to answer this for Extreme Arcade Gaming. However, we identify the most important violations as those that circumvent break competitive gaming. That is, starting, stopping, or modifying a competitive game. We found that an unauthorized user with network access can do all of the aforementioned by accessing an unauthenticated telnet server running on the game server. This policy violation should be addressed first.

The administrator password, TLS certificate, and RSA private key are hardcoded for every MicroCade gaming server. This also means that every gaming server has the same secrets. This hardcoding enables an attacker who successfully reverse engineers the MicroCade software to spoof a legitimate MicroCade and obtain unauthorized access.

As we previously mentioned, considering the severity of a policy violation can be difficult. If the policy doesn't directly impact patient safety, we have to consider how it might enable some other violation that impacts patient safety. To help illustrate this, we have included an attack tree. An attack tree shows security failures along with the flow of events that lead to the failure. We provide an attack tree for the MicroCade in the Figure following.

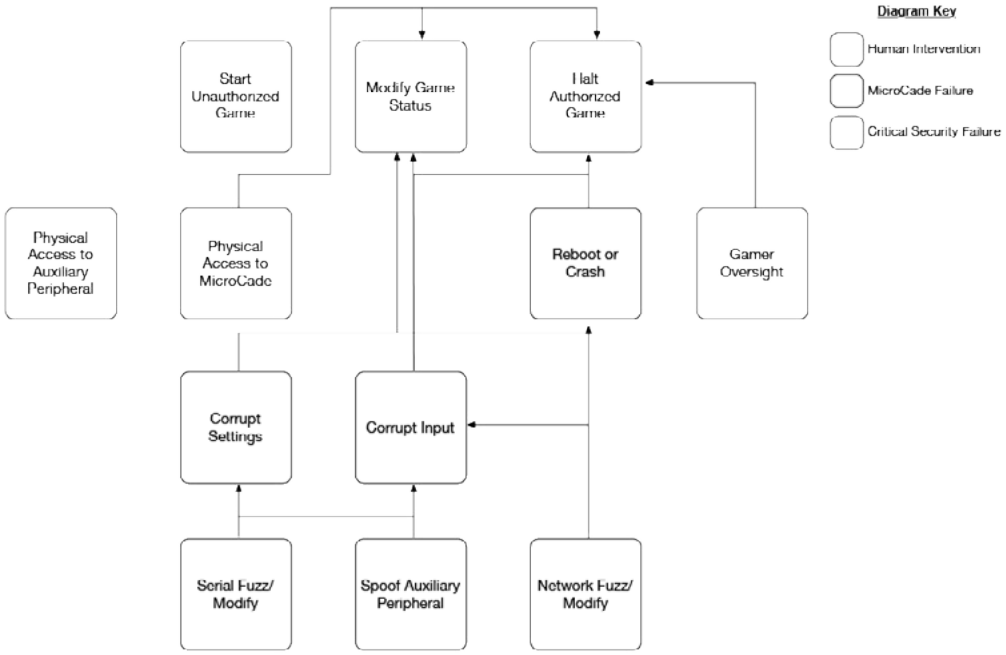

Figure X: MicroCade attack tree.

In the attack tree depicted above, we represent critical security failures as red nodes at the top of the tree (technically a forest). Working backwards, we depict failures in the MicroCade and related systems. Green nodes are those that require human intervention with physical access. We note that this tree is theoretical.

Attack trees are useful, not only in evaluating the severity of an attack, but it also for identifying potential future attack vectors. Attack trees can also be used in analyzing the "design/implementation" divide of security failures. That is, some problems are inherent in the design; in other cases, the design is fundamentally sound but one or more components implemented incorrectly still lead to a compromise.

A. Proposed Solutions

It is important that secrets stored on the MicroCade not be hardcoded. The threat of leaking a global secret is greatly reduced by dynamically generating new secrets for each individual gaming server. Cryptographic hardware can be used to further protect secrets such as the RSA private key. Also, insecure network protocols should not be used. While telnet can enforce authentication, credentials are sent unencrypted and thus recoverable by a passive network eavesdropper. We recommend secure protocols such as SSH with credentials that are, again, not hardcoded.

Extreme Arcade Gaming shared with us their preliminary plan for mitigating the vulnerabilities and weaknesses that we have identified. Their plan includes the replacement of the telnet server with SSH. In particular, Extreme Arcade Gaming is following RFC4256 to implement one-time passwords. Extreme Arcade Gaming's plan also includes the generation of system secrets such as encryption keys on system setup. These secrets will be stored in a PCI-e based hardware security module.

We agree and support Extreme Arcade Gaming's preliminary plan for mitigating the vulnerabilities and weaknesses that we have identified. Also, we recommend the following:

- Auxiliary peripherals should also generate their own unique secrets.
- The peripherals and MicroCade gaming server should be mutually authenticated.

{We might also provide a hardening guide if it's a commodity OS… checklist with todos.}

B. Additional Comments

Milestone: In accordance with the approved SOW, upon acceptance by Extreme Arcade Gaming, this document concludes the review.

No-warranty Reminder: Cybersecurity is based on best practice risk management techniques. No computer network is impervious to assault and completely secure. No amount of testing can discover all possible vulnerabilities. This security assessment does not guarantee the complete security of any computer system, including those cited in this letter. Harbor Labs hereby disclaims responsibility for, and shall not be liable for claims, losses or damages resulting from the penetration of Extreme Arcade Gaming products.

VI. Security Governance

Per the customer's request, the following set of security activities are proposed for adoption as industry best practices for the ongoing maintenance of the client's security posture. These activities are intended to enable strategic oversight of internal security practices, including management roles and responsibilities, establishing communication channels, identifying legal issues, risks and impact, and applying these activities to the client's business objectives. Should the client choose to engage further, Harbor Labs can serve as a strategic cybersecurity partner in support of each of the following activities.

1. Penetration testing
    a. Requirements
    b. Qualifications
    c. Recommendations
        i. White-box vs. black-box penetration testing

ii. Repeat penetration testing for every major release

2. Internal Organization
   a. Cybersecurity lead position
      i. Qualifications, skills, and expertise
   b. Relevant Academic and industry conferences
   c. Design and Architecture Documents
   d. Standards and regulatory compliance
   e. Internal risk/security management processes
   f. Proactive and Retroactive security policies
   i. Response plan
   g. Tradeoffs with safety, usability, reliability, and supportability
3. Product development
   a. Security steps performed during lifecycle of system
      i. Design
      ii. Development
      iii. Q/A
      iv. Manufacturing
      v. Commercial Launch
      vi. Updating and long-term support

References

sample report

harborlabs
CYBER . SCIENCE

About Harbor Labs

Harbor Labs has over a decade of experience in software and networking consulting. Our staff includes PhD-level researchers with a background in medical device security and whom employ an academic approach to performing security-related analysis and technical review.

Some of our past projects include:

- Medical Device Penetration Testing. We evaluated the security of multiple infusion pump products and their related systems. This effort required us to be intimately familiar with the hardware, software, and protocol design and implementation for each product. We gained this familiarity via a white-box approach. Specifically, we acquired internal documentation (i.e., requirements, design, architecture, specifications, data sheets, roadmaps, data and process flow documents) and we established a relationship with the manufacturer's engineers and project managers (e.g., architects, programmers, hardware designers). We also setup a lab environment consisting of an isolated network, infusion pumps, and related systems. These sources of information and test environment allowed us to create an encompassing list of security policies to begin our penetration testing with. Upon completing our tests, we wrote a detailed report that included the security policies, threats, vulnerabilities, attacker model (goals, capabilities, and relation to the system), and whether a given security policy was violated or not. We included technical appendices that further detailed our test results and provided all customer code we wrote so that the manufacturer may reproduce our results. Lastly, we provided recommendations to the manufacturer to address the issues we had identified.

- Design and Development of Security Components and Libraries. For this multi-year project, we developed a wide range of security-related components for a client product. Early deliverables included an encryption library, which was subsequently FIPS-140 certified. Later, we designed and implemented more advanced extensions to this library such as variants that used graphics cards for hardware acceleration, and a prototype secure file-system. One of our final projects was the design, implementation, and testing of a complete secure-communications system.

- Security Evaluation of Client Products. We evaluated the security of a wide range of products for many clients. Sometimes our clients requested reviews at the design stage only, and our investigation focused on documents and figures. Other times, we were asked to conduct "black box" reviews on released or soon-to-be released products. Combined, we have analyzed protocol designs, system designs, source code, and hardware for many systems.

- Source Code Analysis. In addition to security-based evaluations, we have conducted evaluations of source code for additional purposes. We have, for example, compared source code between two companies while investigating accusations of code theft, reviewed systems to determine to what extent the company was protecting sensitive information, and different versions of the same product to identify how the system has changed and evolved over its lifetime. From these numerous reviews, we have examined

harborlabs
CYBER . SCIENCE

code from many languages include C, C++, x86 Assembly, Python, Perl, PL/SQL, Java, C#, JavaScript, PHP, and others.

- Privacy Evaluation. Various copyright holders with digital piracy concerns created a joint organization known as the Center for Copyright Information (CCI). At the request of CCI, we evaluated their methods and practices for investigating piracy to ensure that consumer privacy was sufficiently protected.

Appendix A: Comments Matrix

The contents of this matrix are reproduced as is from document [citation]. The tone is informal between us and Extreme Arcade Gaming engineers.

#	Google Doc	Page #	Harbor Labs Question or Comment
1		1	

About Harbor Labs

Harbor Labs is a leading provider of cyber science consulting services, specializing in cryptography,, network security audits, software vulnerability assessments, and secure programming. Our elite staff of cyberscientists comprises many of the industry's foremost experts in their respective cyber disciplines and are among the first to be contacted when a high-profile, national-level cyber event occurs.

info@harborlabs.com
1-833-CYBR SCI
106 Old Court Rd,
Suite 305, Pikesville,
MD 21208