SOFTWARE REQUIREMENTS SPECIFICATION

for

FreeEDR

Version 3.0.0 approved

Prepared by:
Bryan Bolesta
Ryan Fiers
Declan Kelly
Matthew Horger
Layla Phills
Zachary Santoro
Marisa Tranchitella

Team: FreeEDR
November 3, 2019

Contents

Revision History

Name	Date	Reason For Changes	Version
1.0.0	18–10–19	Initial Structure	mhorger
1.1.0	21–10–19	Design Constrains	dkelly, bbolesta
1.1.1	24–10–19	User Characteristics	lphills
2.0.0	27–10–19	Draft Submission	mhorger
3.0.0	03–11–19	Final Submission	mhorger

1 Introduction

1.1 Purpose of Document

The purpose of this software requirement specification document is to outline the functionality and features of our proposed project. The end goal of this document is to assure that the intended audience of FreeEDR understands how FreeEDR will perform, maintained, and potentially developed further. This document will not describe how the system works in detail or how infrastructure teams should utilize the project. The intended audience of this document is three-fold: first is infrastructure team members of an organization (end-users) interested in implementing our system to enhance their security footprint of workstations. Second is developers who can review the project's requirements to see where time and research efforts can be utilized to improve the project. Lastly is UAT testers who can test our product when deployed in an environment to find potential security risks or unexpected bugs that stem from requirements.

1.2 Project Scope

FreeEDR is an open-source endpoint detection and response system for small infrastructure teams and organizations to utilize free of charge. FreeEDR will be completed by June 2020 for a total time allocation of 9 months. Continuous improvements and support of this project will be assumed by Security Risk Advisors, located at 1760 Market Street in Philadelphia, PA after the completion of the initial deployment/release phase.

1.3 Definitions, Acronyms, and Abbreviations

1. AD: active directory, a package of special services to manage permissions and resources on Windows workstations
2. API: application programming interface, technology used for transmitting data between sources such as clients, servers, databases, etc.
3. EDR: also known as endpoint detection and response, a technology used to address the needs for continuous coverage against advanced threats.
4. GPO: group policy object, used when policy settings need to apply to multiple Windows workstations
5. Sigma: generic, open signature format that allows relevant log events to be reported straightforward

6. SigmaC: tool used to translate Sigma format rules to the language of choice
7. SIEM: Security Information and Event Management
8. SIRT: Security Incident Response Team
9. SOC1: also known as a system and organization controls report, used for making sure an organization's internal control procedures are being properly followed.
10. Threat Intelligence Sources: security feeds from vendors, government / public, and private sources that provide information about known IT vulnerabilities and risks for organizations.
11. QA: Quality Assurance

1.4 Overview

Organizations that don't have a significant security budget can find it difficult to include workstations in their monitoring scope. Forwarding logs from all the organization's workstations can be expensive because most SIEMs are priced based on log ingestion and tools such as EDR are just as expensive. This project aims to setup a series of scripts which will allow organizations who don't have the ability to purchase or implement an enterprise solution to monitor workstation traffic on a domain network.

2 Overall Description

2.1 Product Functions

1. Generate numerous security alerts against internal network traffic where FreeEDR is deployed in.
2. Deploy a secure repository which can update, deliver, and maintain rules that produce said security alerts above.
3. Allow for internal users to create customizable rules and network processes that are stored on the mentioned secured repository.
4. Establish architecture for sending automatic email and message alerts when security alerts are activated within FreeEDR.
5. Provide an interactive dashboard in order to produce reports, logs, and other auditable information detailing specific time-stamped information.

2.2 User Characteristics

2.2.1 Client

This user is responsible for having their workstation's network traffic monitored in an organization where FreeEDR is deployed at. This user should have no privileges to modify or change any aspect of FreeEDR's deployment in an environment.

2.2.2 Script Manager

This user is responsible for the retainment, management, and deployment of the provided PowerShell scripts. The script manager must have the privileges and means to deploy and run the scripts on Windows endpoints across the organization. The main function of this user is to deploy the series of scripts to any in-scope endpoints, retain the scripts, and re-deploy scripts when necessary.

2.2.3 Incident Response Manager

This user is responsible for ensuring that alerting methods are set-up and properly configured. This user is also responsible for reviewing, responding, and analyzing security alerts. The Incident Response Manager must have view access to the platform that the alerts are being sent (SIEM, Email, or Message). The main function of the Incident

Response Manager is to review and respond to the security alerts generated by the FreeEDR scripts.

2.2.4 Incident Response Supporter

This user is responsible for aiding the Incident Response Manager in responding to incidents in real time. This user must have the privileges and means to view the reports curated specifically for less technical people. The main function of the incident response supporter is to follow basic instructions to complete tasks that will offset the responsibilities of the Incident Response Manager during an incident response.

2.2.5 System Auditor

This user is responsible for auditing the functionality and use of the system. The auditor must have view access of all system and user logs. This user must also have access to any previous audit reports. The main function of this user is to ensure that best practices were followed, the system is being used as effectively as possible and to report any compliance issues discovered to the appropriate parties.

2.2.6 Dashboard Infrastructure Manager

This user is responsible for maintaining and deploying future releases of the Audit Dashboard for auditors to view. The infrastructure manager must make sure that the dashboard is consistent with the information that is being produced for the system auditor, and have continuous support capacities if some errors were to occur with the reporting tool.

3 Specific Requirements

3.1 Functional Requirements

R1. Server - Rule Storage

- R1.1. FreeEDR will allow for script managers to store correlation logic for process and network events.
- R1.2. FreeEDR will allow read access to the contents of the server for Incident Response Managers to monitor deployments to the secured repository.
- R1.3. FreeEDR will allow for server to communicate and connect with threat intelligence sources to discover new correlation rules.
- R1.4. FreeEDR will grant security access to the organization's system administrator to update correlation logic on the server.
- R1.5. FreeEDR will restrict read access from unauthorized clients in the organization who have not received internal approval to view the server.

R2. Client - Rule Processing

- R2.1. FreeEDR will establish a communication channel between clients and the Rule Storage Server to pull down correlation rules.
- R2.2. FreeEDR must restrict clients from writing to the rules repository.
- R2.3. FreeEDR will force clients to check correlation rules against the event log every hour. This process should be non-disruptive to normal client activity.
- R2.4. FreeEDR will establish network security protocols to permit clients to communicate with APIs.
- R2.5. FreeEDR will allow clients to receive information from APIs to perform network and process forensics on an event.
- R2.6. FreeEDR must be able to store clients' process and network forensic information within the organization's filesystem.
- R2.7. FreeEDR will store the referenced forensic information for an amount of time set by the organization's system administrator.
- R2.8. FreeEDR must forward forensic event from clients to the Incident Response Manager in order for them to properly perform their user roles.
- R2.9. FreeEDR will be able to distinguish which forensics must be applied for different client process and network events.

R3. Dashboard

R3.1. FreeEDR must allow for communication between the dashboard and the secured repository used for rule storage.

R3.2. FreeEDR must permission all users read access to the deployed dashboard. FreeEDR must block all external traffic to the dashboard.

R3.3. FreeEDR must provide the ability for clients to view previously generated reports as well as produce fresh reports.

R3.4. FreeEDR must allow for the Dashboard Infrastructure Manager to permission certain clients to view specific reports and actions.

R3.5. FreeEDR will allow clients to select a range of dates for report generation within the dashboard.

R3.6. FreeEDR will allow clients to select a specific format to download their generated reports from the dashboard.

R3.7. FreeEDR must have an option to export a dashboard report to send via interdepartmental communication (email, IM, etc).

R3.8. FreeEDR will allow for the dashboard to have a responsive algorithm that allows for regeneration of reports once fresh data is produced. Please see Data Requirements further on the data types.

R3.9. FreeEDR must maintain the dashboard so that it is accessible 90% of normal business hours, with the exception being disaster recovery downtime / failover procedures.

R4. Data

R4.1. FreeEDR will have an established process in order to track requests, actions, etc in regards to manipulating data in the system.

R4.2. FreeEDR will present Incident Response teams with data on endpoint events (i.e. registry modifications, cross-process events, file executions, network connections).

R4.3. FreeEDR must have 100% uptime access to relevant data sources needed for operations.

R4.4. FreeEDR must keep data for up to 5 years in order to comply with SOC1 reporting / audit procedures. Data past 5 years is outside the jurisdiction of auditable actions and can be disposed.

R4.5. Data transmitted via every API in FreeEDR must be under the proper protocols for security (POST) and sensitive information must be encrypted before transit.

R4.6. FreeEDR should allow the dashboard to access all necessary data in order to produce reports. This data includes forensic event information, user machine configuration, and standard log outputs.

R4.7. FreeEDR must display data in the dashboard in a concise, readable format with an option for details to be viewed separately.

3.2 Non-Functional Requirements

N1. Hardware

N1.1. FreeEDR will only use hardware in compliance with Security Risk Advisors minimum security requirements.

N1.2. FreeEDR will be deployed on a secure and segmented part of the Drexel CyberDragon's server until system ownership is transferred to Security Risk Advisors.

N2. Network

N2.1. FreeEDR will not use or interact with the Security Risk Advisors network in any way that violates any of their privacy policies.

N2.2. FreeEDR will have 98.9% availability on the network during standard business hours.

N2.3. FreeEDR will return a response within 15 seconds of the client's request, with larger data requests returning within 5 minutes of the request.

N3. Deployment

N3.1. FreeEDR will be able to handle all exceptions created by erroneous user input.

N3.2. All deployed versions of FreeEDR must have passed all QA /UAT tests.

N3.3. FreeEDR will consume at most 250 megabytes of memory.

N3.4. FreeEDR's Mean Time to Change (MTTC) for issues will be ¡ 3 person days.

N4. Other

N4.1. FreeEDR will provide training information to clients interested in being able to identify the difference between process events and network events.

N4.2. FreeEDR's dashboard must be flexible for modifications in order to add/remove reports as needed by auditors.

3.3 Data Requirements

D1. Event

An event is an object captured and displayed by an Event Viewer. These events are triggered by the rules, which are demonstrated below. A sample event is one that has the following data fields:

D1.1. Event Type

Critical, Error, Warning

D1.2. Event ID

111

D1.3. Source

(Any name of application)

D1.4. Log Location

(path of application)

D1.5. Date and Time

11-03-2019 T12:43:00Z

D1.6. Task Category

(any user defined category)

D1.7. Keywords

(any user defined keyword)

D1.8. Computer

DESKTOP-341k3

D1.9. User

SRAPROD/nascoli

D1.10. OpCode

1

D1.11. More Information

(any details go here)

D2. Rule

A correlation rule is user-defined logic that can be used to trigger security alerts when endpoints are targeted. For the purposes of FreeEDR, certain data fields can be customized in order to provide modified functionality. An example rule can have the following data fields:

D2.1. Title

Suspicious SQL Error Messages

D2.2. Status

experimental

D2.3. Description

Detects SQL error messages that indicate probing for an injection attack

D2.4. Author

Bjoern Kimminich

D2.5. References

http://www.sqlinjection.net/errors

D2.6. Logsource

category: application

product: sql

D2.7. Detetction

keywords

Oracle: quoted string not properly terminated

MySQL: You have an error in your SQL syntax

SQL Server: Unclosed quotation mark

D2.8. Falsepositives

Application bugs

D2.9. Level

high

D3. Rule Repository

A rule repository is a directory structure that can securely store rules in the file format above. Example data structures include:

D3.1. NTFS

D3.2. FAT32

D3.3. Cloud Storage

D4. Network Log

A network log captures the information when any traffic occurs between clients and endpoints. Network logs typically have the following information:

D4.1. Name of Machine

DESKTOP-3412k3

D4.2. User Account

SRAPROD/nascoli

D4.3. Date Time of Request

11-03-2019 T12:43:00Z

D4.4. IP Address

192.141.53.2

D4.5. HTTP Response Status Code

202 OK

D4.6. Headers

Keep-Alive: *

D4.7. Requested IP Address

64.10.343.12

D4.8. Number of Packets Transmitted

333

3.4 Design Constraints

DR1. FreeEDR system architecture must be split in order to establish a server/client relationship for security measures.

DR2. Clients checking for new correlation rules in FreeEDR must be an established fixed interval to allow for appropriate time to gather said rules that have been recently deployed.

DR3. Client process and event information must be stored in a single shared location in order for the FreeEDR's dashboard to know where to access the information.

DR4. FreeEDR's reporting dashboard must be hosted on the same server used for rule storage due to security constraints for processing information.

DR5. Correlation rules should be written in Sigma to allow sharing through threat intelligence platforms such as Threat Alert Logic Repository (TALR)

DR6. FreeEDR correlation rules must be translated to PowerShell Get-Event Queries as this is the supported language in Sigma.

DR7. FreeEDR must be deployed in a Windows environment due to the reliance on Get-Event PowerShell queries.

DR8. FreeEDR must deploy Syson on client workstations to capture the appropriate events for the system to perform properly.

DR9. FreeEDR must satisfy all of Nick Ascoli's, our external stakeholder for this project, UI preferences when system ownership is transferred to Security Risk Advisors.

Bibliography

[1] Beyond Feeds: A Deep Dive Into Threat Intelligence Sources, *Recorded Future*, https://www.recordedfuture.com/threat-intelligence-sources/, 2019.

[2] Threat Alert Logic Repository, *Security Risk Advisors*, https://github.com/SecurityRiskAdvisors/TALR, 2019.