

# LEVANTAMENTO DE REQUISITOS DE DADOS

## 1. INTRODUÇÃO

Para garantir a eficácia da identificação e correção de vulnerabilidade, é essencial compreender as necessidades e expectativas de nossos clientes do Bug HUnter. Esse processo visa a coleta de informações planejadas sobre os sistemas, objetivos de segurança e expectativas de desempenho, permitindo uma abordagem personalizada e eficiente na análise de segurança.

## 2. OBJETIVOS

**Entender as Necessidades e Expectativas do Cliente:** O objetivo principal é compreender as necessidades específicas do cliente em relação à segurança, infraestrutura de rede, ou dispositivos móveis.

**O Escopo dos Testes de segurança:** Definir claramente o escopo da análise de vulnerabilidade, incluindo quais sistemas, plataformas ou componentes devem ser avaliados.

**Identificar Prioridades e Riscos:** Identificar os pontos críticos e mais vulneráveis no ambiente do cliente, com base em sua infraestrutura, processos e objetivos de negócios .

## 3. ESCOPO

### 1. Objetivo Principal

Detectar, analisar e mitigar ameaças cibernéticas em computadores, como:

- Malware (vírus, ransomware, spyware, etc.).
- Vulnerabilidades de software/sistemas
- Configurações inseguras
- Arquivos ou atividades suspeitas

## 2. Funcionalidades Principais

- **Varreduras Antimalware:** Identificação de códigos maliciosos em arquivos, processos e memória.
- **Análise de Vulnerabilidades:** Verificação de sistemas desatualizados, patches ausentes ou falhas conhecidas (CVEs).
- **Monitoramento em tempo real:** Detecção contínua de atividades suspeitas (ex: conexões não autorizadas.)

## 3. Escopo de Atuação

- **Dispositivos Alvo:**
  - Computadores (Windows, macOS, Linux)
  - Servidores (opcional, dependendo do software)
- **Áreas verificadas:**
  - Arquivos locais e redes compartilhadas
  - Registros do sistema (Windows Registry)
  - Processos ativos de serviços em execução
  - Tráfego de rede local (se incluído)

## 4. Limitações (Fora do Escopo)

- **Não é um firewall:** Não bloqueia tráfego malicioso em tempo real (a menos que integrado a um site de segurança)
- **Não substitui atualizações humanas:** Requer intervenção para corrigir vulnerabilidades (ex: instalar patches).
- **Depende de assinaturas:** Pode não detectar ameaças desconhecidas (zero-day) sem atualizações regulares.
- **Escopo físicos:** Não cobre dispositivos externos (ex: smartphones) a menos que especificados.

## **4. REQUISITOS FUNCIONAIS**

### **➤ Cadastro e gerenciamento**

#### **1.Cadastro e Gerenciamento dos clientes**

o sistema deve permitir a cadastro de novos clientes contendo os dados pessoais (nome completo, Email, cpf, numero de telefone) e as informações da empresa e etc. , também haver a opção de poder atualizar as informações dos clientes.

#### **2.Cadastro e gerenciamento de desenvolvedores**

O sistema deve permitir que desenvolvedores se cadastrem no sistema afim de buscar oportunidades dentro da empresa, seria uma forma de buscar oportunidades de trabalho , deve conter (nome, cpf, data de nascimento, numero de telefone, Email) e seu currículo, portfólio e etc. , também permitir alterações nos dados dos desenvolvedores.

#### **3.Gerenciamento dos projetos dos clientes**

o sistema deve permitir com que possamos organizar os projetos dos clientes, de forma com que mostre o nome, data, prazo, tipo de serviço, tendo a possibilidade de alterar os mesmos.

#### **4.consulta das informações**

O sistema deve ter um sistema organizado, podendo consultar qualquer informação dentro dele, de forma otimizada para os administradores poder revisar e buscar as informações.

## **5. REQUISITOS NÃO FUNCIONAIS**

### **1. Segurança**

O sistema deve ter uma proteção de dados , como autenticação em dois fatores para realizar o login, ter diferentes níveis de acesso, como os mais altos tendo todas as informações do sistema apenas para administradores, e o direito de alterar as informações, e os mais baixos, para os clientes e desenvolvedores olharem seus dados e acompanhar seu projetos e atualizar seus dados pessoais

### **2. utilizar criptografia**

Utilizar criptografias SSL/TLS para proteger os graficos

### **3. desempenho**

Deve ser capaz de realizar 5000 cadastros simultaneos, sem comprometer o desempenho e tero tempo maximo de resposta de 2 segundos em situações normais de uso

#### **4. usabilidade**

Deve ter um interface intuitiva tendo acessibilidade a pessoas com necessidades especiais, e com suporte a outros dispositivos

#### **4. Manutenibilidade**

O sistema deve ser capaz de aceitar adição de novas funcionabilidades sem grandes alterações no código, e o código deve ser de fácil manutenção

#### **5. Escalabilidade**

O sistema deve suportar o crescimento de alunos ao longo do tempo

## **6. TECNOLOGIAS**

### **7.1 Front -end.**

- **Frameworks:** YARA, OpenVAS, Zeek, TheHive, Cortex, MISP (Malware Information Sharing Platform)
- **Linguagens:** TypeScript, React, Angular

### **7.2 Back-end**

- **Frameworks:** AWS, Kafka, Elasticsearch
- **Linguagens:** Go, Python, C++

## **7. INTERGRAÇÕES**

### **1- Programa de inovação**

Implementar programas que promete inovações para a cibersegurança.

Treinamento de novos membros com profissionais especializados, por um determinado período.TH

### **2- Definir um meio de gerenciamento de dados e armazenamento**

Criar e gerenciar os dados com cuidado, separando: pagamentos, matrículas, e pedidos em áreas específicas.

## 8. RESTRIÇÕES

° A startup deve cumprir a Lei Geral de Proteção de Dados (LGPD), para definirem os limites para a aquisição e manipulação de dados pessoais e empresariais.

° Implementar medidas de segurança robustas, como os firewalls e antivírus atualizados, backups frequentes e atualizações regulares de software.

° Proteger a rede Wi-Fi, alterando o nome do ponto de acesso sem fio ou roteador (SSID) e usando uma senha de tecla pré-compartilhada complexa (PSK).

## 9. PRAZOS E MARCOS

### Planejamento e Definição de Objetivos

- **Prazo:** 1-2 semanas
- **Marcos:** Definição de expectativas e metas, como o número de vulnerabilidades a serem encontradas.
- Definição de expectativas e metas, como o número de vulnerabilidades a serem encontradas.
- Elaboração de um plano de comunicação com a equipe do cliente.

### Configuração e Preparação do Ambiente

- **Prazo:** 1 semana
- **Marcos:** Configuração de ferramentas e plataformas necessárias para a busca de bugs (como scanners de vulnerabilidade, sistemas de rastreamento de bugs, etc.).
- Treinamento ou alinhamento da equipe de caçadores de bugs com o escopo do projeto.
- Definição de processos de documentação e reporte de falhas encontradas.

### Pesquisa e Testes de Vulnerabilidades (Fase Ativa de Caça)

- **Prazo:** 3-6 semanas (dependendo da complexidade do projeto)
- **Marcos:** Início da exploração e pesquisa de vulnerabilidades, através de testes manuais ou automatizados.

## **Revisão e Correção dos Bugs**

- **Prazo:** 2-3 semanas (pode variar conforme a complexidade dos bugs)
- **Marcos:** Desenvolvimento e implementação de correções pelas equipes de desenvolvimento.
- Realizar testes pós-correções para garantir que os bugs foram realmente corrigidos e que não houve regressão.
- Garantir que todas as correções atendam aos requisitos de segurança e funcionabilidade.

## **Testes de Validação e Avaliação Final**

- **Prazo:** 1-2 semanas
- **Marcos:** Realizar uma rodada final de testes de validação para verificar se os bugs corrigidos não afetam a estabilidade do sistema.
- Avaliar o desempenho do sistema após as correções e verificar se os requisitos de segurança foram atendidos.
- Revisão final do relatório de bugs e resultados dos testes.

## **Entrega do Relatório Final**

- **Prazo:** 1 semana
- **Marcos:** Realizar uma rodada final de testes de validação para verificar se os bugs corrigidos não afetam a estabilidade do sistema.
- Avaliar o desempenho do sistema após as correções e verificar se os requisitos de segurança foram atendidos.
- Revisão final do relatório de bugs e resultados dos testes.

## 10. Riscos

- **Vazamento de Dados Sensíveis:** Durante a identificação de vulnerabilidades, os bugs Hunter podem acessar informações confidenciais. Se esses dados forem expostos acidentalmente ou mal utilizados, isso pode resultar em sérias consequências legais e danos à reputação da empresa.
- **Responsabilidade Legal:** A atuação em sistemas sem autorização explícita pode levar a problemas legais. É fundamental que os bugs Hunter operem dentro dos limites legais e tenham acordos claros com as empresas para evitar ações judiciais.
- **Dependência de Ferramentas e Tecnologias:** A eficácia de um bug Hunter muitas vezes depende de ferramentas e tecnologias específicas. Se essas ferramentas falharem ou não forem atualizadas, pode haver um aumento no risco de não detectar vulnerabilidades críticas, comprometendo a segurança do sistema.

## 11. Considerações Finais

### Aprovação e Assinaturas

Função	Nome	Assinatura	Data
Gerente de projeto	Victor	Victorsl	04/04/2025
Analista de requisitos	Raphael Rodrigo	Raphael.r Rodrigo.s	04/04/2025
Desenvolvedor Líder	João Victor	JVsave...	04/04/2025
Representante do cliente	Marcelinhos	MarcelinhoAlmeida ...	04/04/2025
Diretor de tecnologia	Bruno	Bruno.b	04/04/2025