代码安全审计报告

共发现 11 个安全问题

dns_proxy.go:18

路径遍历 (Path Traversal) (MEDIUM)

在parse函数中使用os.Open直接打开固定文件名'proxy.config'，未进行路径合法性校验。攻击者可能通过符号链接或控制父目录的方式加载恶意配置文件

修复建议： 使用os.Stat检查文件权限和属性，确保为常规文件；添加路径校验逻辑限制文件必须位于指定安全目录下

dns_proxy.go:57

拒绝服务 (Denial of Service) (MEDIUM)

DNS请求处理未设置超时控制，当上游DNS响应缓慢时会导致goroutine堆积，消耗系统资源

修复建议： 在dns.Exchange调用时添加context.WithTimeout设置合理超时时间，例如5秒

dns_proxy.go:65

配置重载竞态条件 (Race Condition) (HIGH)

SIGUSR1信号处理中，recordLock采用完全锁导致配置重载期间服务不可用。长时间文件解析可能引发服务中断

修复建议： 采用双缓冲机制：维护新旧两个配置副本，原子切换指针减少锁粒度；或使用sync.RWMutex的读写锁优化

dns_proxy.go:24

敏感信息泄露 (LOW)

日志输出完整域名映射关系，可能暴露内部网络拓扑等敏感信息

修复建议： 在生产环境关闭调试日志输出，或对日志中的敏感信息进行脱敏处理

dns_proxy.go:44

DNS缓存投毒 (DNS Cache Poisoning) (HIGH)

未验证上游DNS响应的事务ID、源端口等字段，攻击者可伪造响应进行缓存投毒

修复建议： 实现DNSSEC验证，检查响应与请求的匹配性（包括Question段、ID等），使用随机源端口

subdomain_guesser.go:103

DNS 查询缺乏速率限制 (MEDIUM)

代码中未对DNS查询进行速率限制，可能导致大量并发请求被发送到DNS服务器（默认使用8.8.8.8:53），可能触发服务器的速率限制或被视为拒绝服务攻击（DoS）。

修复建议： 在worker函数中添加延迟控制（如使用time.Sleep）或限制并发查询的数量，避免短时间内发送过多请求。

subdomain_guesser.go:85

潜在资源耗尽 (MEDIUM)

使用无缓冲的通道（tracker, gather）和大量goroutine可能导致资源竞争或阻塞，特别是在处理大字典时可能耗尽内存或文件句柄。

修复建议：优化通道缓冲机制，确保及时关闭和释放资源。使用sync.WaitGroup替代tracker通道来同步goroutine。

subdomain_guesser.go:112

错误处理不完善 (LOW)

在main函数中打开文件时直接调用panic，且未检查scanner.Err()，可能导致潜在的文件读取错误被忽略。

修复建议：使用更友好的错误处理代替panic，并在扫描后检查scanner.Err()以处理读取期间的错误。

subdomain_guesser.go:19

缺乏超时控制 (MEDIUM)

DNS查询（dns.Exchange）未设置超时时间，默认可能长时间阻塞，导致程序停滞或资源无法释放。

修复建议：在dns.Exchange调用时设置合理的超时时间，例如使用context.WithTimeout或自定义dns.Client结构体指定超时。

字符串逃逸.php:3

Session Injection (HIGH)

未经验证的用户输入直接存储在会话变量中，可能导致会话数据篡改或恶意数据注入。

修复建议：应对$_GET['ben']进行严格的输入验证和过滤，例如使用白名单验证或适当的清理函数。

字符串逃逸.php:3

Cross-Site Scripting (XSS) (MEDIUM)

用户提供的输入未经处理直接存储到会话中，若后续在页面展示时未转义，可能导致XSS攻击。

修复建议：在将会话数据输出到HTML前，使用htmlspecialchars()等函数进行输出编码。