



(12) 发明专利申请

(10) 申请公布号 CN 106909469 A

(43) 申请公布日 2017. 06. 30

(21) 申请号 201510970585. 4

(22) 申请日 2015. 12. 22

(71) 申请人 中国移动通信集团公司
地址 100032 北京市西城区金融大街 29 号

(72) 发明人 石松泉 左敏 刘芳 耿慧拯

(74) 专利代理机构 北京派特恩知识产权代理有限公司 11270

代理人 张振伟 张颖玲

(51) Int. Cl.
G06F 11/10(2006. 01)

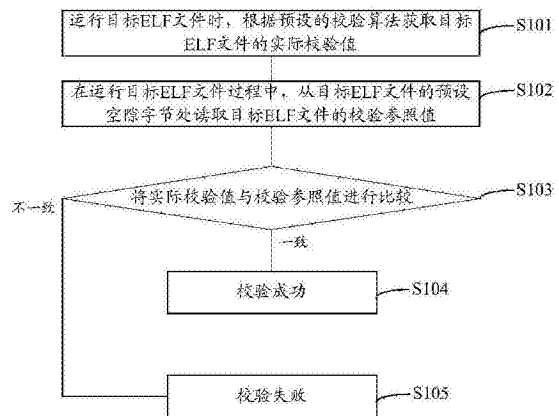
权利要求书2页 说明书6页 附图4页

(54) 发明名称

一种文件校验方法和装置

(57) 摘要

本发明实施例公开了一种文件校验方法和装置,该方法包括:运行目标 ELF 文件时,根据预设的校验算法获取所述目标 ELF 文件的实际校验值;在所述运行目标 ELF 文件过程中,从所述目标 ELF 文件的预设空隙字节处读取所述目标 ELF 文件的校验参照值;将所述实际校验值与所述校验参照值进行比较;当所述实际校验值与所述校验参照值一致时,确定校验成功;当所述实际校验值与所述校验参照值不一致时,确定校验失败。从而无需对 ELF 文件中的字符进行替换,在 ELF 文件运行过程中,通过预先保存的校验值就能够对自身进行校验,不占用额外的存储空间。



1. 一种文件校验方法,其特征在于,所述方法包括:

运行目标可执行链接格式ELF文件时,根据预设的校验算法获取所述目标ELF文件的实际校验值;

在所述运行目标ELF文件过程中,从所述目标ELF文件的预设空隙字节处读取所述目标ELF文件的校验参照值;

将所述实际校验值与所述校验参照值进行比较;

当所述实际校验值与所述校验参照值一致时,确定校验成功;

当所述实际校验值与所述校验参照值不一致时,确定校验失败。

2. 根据权利要求1所述的方法,其特征在于,所述目标ELF文件的预设空隙字节具体包括:所述目标ELF文件中各节之间的空隙以及所述目标ELF文件各段的预留保留字段。

3. 根据权利要求2所述的方法,其特征在于,在所述根据预设的校验算法获取所述目标ELF文件的实际校验值之前,所述方法还包括:

遍历所述目标ELF文件,获取所述目标ELF文件中的空隙字节;

按照预设的写入规则将所述目标ELF文件的校验参照值写入空隙字节。

4. 根据权利要求3所述的方法,其特征在于,所述遍历所述目标ELF文件,获取所述目标ELF文件中的空隙字节,具体包括:

根据所述目标ELF文件中的ELF Header字段信息获取所述目标ELF文件各段的保留字段地址;以及,

根据所述目标ELF文件中节头部表Section Header Table获取所述目标ELF文件中各节之间的空隙。

5. 根据权利要求3所述的方法,其特征在于,所述从目标ELF文件的预设空隙字节处读取所述目标ELF文件的校验参照值,具体包括:

获取所述目标ELF文件的预设空隙字节地址;

从获取得到的预设空隙字节地址中读取所述目标ELF文件的校验参照值。

6. 一种文件校验装置,其特征在于,所述装置包括:运行单元、获取单元、读取单元和校验单元;其中,

所述运行单元,用于运行目标可执行链接格式ELF文件;

所述获取单元,用于所述运行单元运行目标ELF文件时,根据预设的校验算法获取所述目标ELF文件的实际校验值;

所述读取单元,用于所述运行单元运行目标ELF文件过程中,从所述目标ELF文件的预设空隙字节处读取所述目标ELF文件的校验参照值;

所述校验单元,用于将所述实际校验值与所述校验参照值进行比较;以及,

当所述实际校验值与所述校验参照值一致时,确定校验成功;以及,

当所述实际校验值与所述校验参照值不一致时,确定校验失败。

7. 根据权利要求6所述的装置,其特征在于,所述目标ELF文件的预设空隙字节具体包括:所述目标ELF文件中各节之间的空隙以及所述目标ELF文件各段的预留保留字段。

8. 根据权利要求7所述的装置,其特征在于,所述装置还包括遍历单元和写入单元;其中,

所述遍历单元,用于遍历所述目标ELF文件,获取所述目标ELF文件中的空隙字节;

所述写入单元,用于按照预设的写入规则将所述目标ELF文件的校验参照值写入所述遍历单元获取到的空隙字节。

9.根据权利要求8所述的装置,其特征在于,所述获取单元,具体用于:

根据所述目标ELF文件中的ELF Header字段信息获取所述目标ELF文件各段的保留字段地址;以及,

根据所述目标ELF文件中节头部表Section Header Table获取所述目标ELF文件中各节之间的空隙。

10.根据权利要求8所述的装置,其特征在于,所述读取单元,具体用于:

获取所述目标ELF文件的预设空隙字节地址;

从获取得到的预设空隙字节地址中读取所述目标ELF文件的校验参照值。

一种文件校验方法和装置

技术领域

[0001] 本发明涉及信息安全技术,尤其涉及一种文件校验方法和装置。

背景技术

[0002] 可执行链接格式(ELF,Executable and Linking Format)文件是一种二进制文件,可以被调入中央处理器中直接运行,也可以与多个ELF文件连接在一起而形成一个可执行文件,因此,被广泛地应用于电脑、嵌入式系统及移动终端等应用程序中。

[0003] 当前,为了防止第三方通过反编译工具对ELF文件进行破解,出现了很多对ELF文件进行保护的方案,比如,将ELF文件符号表中的符号用加密符进行替换,使得加密后的符号难以阅读,提高文件被恶意破解的门槛,从而对ELF文件进行保护。

[0004] 但是上述方案在对ELF文件解密时,需要对保护后的ELF文件中的加密后的符号进行解密,因此,在具体实现时,不仅需要额外对解密所使用的字典进行保存,还需要另外的程序对保护后的ELF文件进行解密,占用额外的存储空间。

发明内容

[0005] 为解决上述技术问题,本发明实施例期望提供一种文件校验方法和装置,无需对ELF文件中的字符进行替换,ELF文件在运行时就能够实现对自身的校验,不占用额外的存储空间。

[0006] 本发明的技术方案是这样实现的:

[0007] 第一方面,本发明实施例提供了一种文件校验方法,该方法可以包括:

[0008] 运行目标可执行链接格式ELF文件时,根据预设的校验算法获取所述目标ELF文件的实际校验值;

[0009] 在所述运行目标ELF文件过程中,从所述目标ELF文件的预设空隙字节处读取所述目标ELF文件的校验参照值;

[0010] 将所述实际校验值与所述校验参照值进行比较;

[0011] 当所述实际校验值与所述校验参照值一致时,确定校验成功;

[0012] 当所述实际校验值与所述校验参照值不一致时,确定校验失败。

[0013] 在上述方案中,所述目标ELF文件的预设空隙字节具体包括:所述目标ELF文件中各节之间的空隙以及所述目标ELF文件各段的预留保留字段。

[0014] 在上述方案中,在所述根据预设的校验算法获取所述目标ELF文件的实际校验值之前,所述方法还包括:

[0015] 遍历所述目标ELF文件,获取所述目标ELF文件中的空隙字节;

[0016] 按照预设的写入规则将所述目标ELF文件的校验参照值写入空隙字节。

[0017] 在上述方案中,所述遍历所述目标ELF文件,获取所述目标ELF文件中的空隙字节,具体包括:

[0018] 根据所述目标ELF文件中的ELF Header字段信息获取所述目标ELF文件各段的保

留字段地址;以及,

[0019] 根据所述目标ELF文件中节头部表Section Header Table获取所述目标ELF文件中各节之间的空隙。

[0020] 在上述方案中,所述从目标ELF文件的预设空隙字节处读取所述目标ELF文件的校验参照值,具体包括:

[0021] 获取所述目标ELF文件的预设空隙字节地址;

[0022] 从获取得到的预设空隙字节地址中读取所述目标ELF文件的校验参照值。

[0023] 第二方面,本发明实施例提供了一种文件校验装置,该装置可以包括:运行单元、获取单元、读取单元和校验单元;其中,

[0024] 所述运行单元,用于运行目标可执行链接格式ELF文件;

[0025] 所述获取单元,用于所述运行单元运行目标ELF文件时,根据预设的校验算法获取所述目标ELF文件的实际校验值;

[0026] 所述读取单元,用于所述运行单元运行目标ELF文件过程中,从所述目标ELF文件的预设空隙字节处读取所述目标ELF文件的校验参照值;

[0027] 所述校验单元,用于将所述实际校验值与所述校验参照值进行比较;以及,

[0028] 当所述实际校验值与所述校验参照值一致时,确定校验成功;以及,

[0029] 当所述实际校验值与所述校验参照值不一致时,确定校验失败。

[0030] 在上述方案中,所述目标ELF文件的预设空隙字节具体包括:所述目标ELF文件中各节之间的空隙以及所述目标ELF文件各段的预留保留字段。

[0031] 在上述方案中,所述装置还包括遍历单元和写入单元;其中,

[0032] 所述遍历单元,用于遍历所述目标ELF文件,获取所述目标ELF文件中的空隙字节;

[0033] 所述写入单元,用于按照预设的写入规则将所述目标ELF文件的校验参照值写入所述遍历单元获取到的空隙字节。

[0034] 在上述方案中,所述获取单元,具体用于:

[0035] 根据所述目标ELF文件中的ELF Header字段信息获取所述目标ELF文件各段的保留字段地址;以及,

[0036] 根据所述目标ELF文件中节头部表Section Header Table获取所述目标ELF文件中各节之间的空隙。

[0037] 在上述方案中,所述读取单元,具体用于:

[0038] 获取所述目标ELF文件的预设空隙字节地址;

[0039] 从获取得到的预设空隙字节地址中读取所述目标ELF文件的校验参照值。

[0040] 本发明实施例提供了一种文件校验方法和装置,通过在ELF文件预留的空隙字节中保存自身的校验值,从而无需对ELF文件中的字符进行替换,在ELF文件运行过程中,通过预先保存的校验值就能够对自身进行校验,不占用额外的存储空间。

附图说明

[0041] 图1为本发明实施例提供了一种文件校验的方法流程示意图;

[0042] 图2为本发明实施例提供了一种ELF文件的整体格式示意图;

[0043] 图3为本发明实施例提供了一种ELF文件的具体内容示意图;

- [0044] 图4为本发明实施例提供了一种将校验参考值写入空隙字节的过程示意图；
- [0045] 图5为本发明实施例提供了一种ELF文件中ELF Header字段信息的结构示意图；
- [0046] 图6为本发明实施例提供了一种ELF文件中节头部表中表项结构示意图；
- [0047] 图7为本发明实施例提供了一种ELF文件中节头部表中表项内容示意图；
- [0048] 图8为本发明实施例提供了一种从空隙字节处读取校验参照值的流程示意图；
- [0049] 图9为本发明实施例提供了一种文件校验装置结构示意图；
- [0050] 图10为本发明实施例提供的另一种文件校验装置结构示意图。

具体实施方式

[0051] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述。

[0052] 实施例一

[0053] 参见图1,其示出了本发明实施例提供了一种文件校验的方法,该方法可以应用于能够运行目标ELF文件的装置,该方法可以包括:

[0054] S101:运行目标ELF文件时,根据预设的校验算法获取目标ELF文件的实际校验值;

[0055] S102:在运行目标ELF文件过程中,从目标ELF文件的预设空隙字节处读取目标ELF文件的校验参照值;

[0056] S103:将实际校验值与校验参照值进行比较;

[0057] S104:当实际校验值与校验参照值一致时,确定校验成功;

[0058] S105:当实际校验值与校验参照值不一致时,确定校验失败。

[0059] 需要说明的是,预设的校验算法可以是消息摘要算法第五版(MD5, Message Digest Algorithm)或安全哈希算法(SHA1, Secure Hash Algorithm)等用于进行数据校验的算法。相应地,实施例中的实际校验值和校验参照值均是和预设的校验算法相对应的校验值。

[0060] 并且,当校验成功后,将继续运行目标ELF文件;当校验失败后,将停止运行目标ELF文件并退出。

[0061] 可以理解地,本发明实施例对步骤S101和步骤S102的执行顺序不进行限定,可以理解地,获取目标ELF文件的实际校验值与读取目标ELF文件的校验参考值之间的先后顺序可以按照运行目标ELF文件的具体运行顺序进行排列,本发明实施例对此不作限定和赘述。

[0062] 由上可知,图1所示的技术方案能够在运行ELF文件时,从ELF文件中的空隙字节处读取得到预存的用于文件校验的参照值,因此,能够在ELF文件运行过程中对自身进行校验,并且无需额外的存储空间占用。

[0063] 实施例二

[0064] 结合图1所示的技术方案,如图2所示的ELF文件的整体格式,ELF文件是一种基于段的数据结构。对于ELF文件来说,文件开始是ELF Header字段信息;随后是程序头部表(Program Header Table);接着是各节Segment的内容信息;然后是节头部表(Section Header Table),节头部表中有字段记录每节Segment的实际长度等信息。节头部表的位置可由ELF Header字段中的值进行指示。为了提高CPU的存取速度,ELF文件中的各节Segment长度会有大小对齐的要求。因此各节Segment数据的实际长度通常都小于等于节Segment大

小,这就产生了节与节之间的空隙,另外,在ELF文件格式中,各段均有预留的保留字段不被使用,因此,目标ELF文件的预设空隙字节具体可以包括:目标ELF文件中各节之间的空隙以及目标ELF文件各段的预留保留字段。

[0065] 如图3所示的一个ELF文件的具体内容示意图,如图3中实线所示划分为三个区域,左区域为地址索引;中间区域为每个地址存储字符所对应的十六进制码;右区域为每个地址存储字符对应的美国标准信息交换代码(ASCII,American Standard Code for Information Interchange)。

[0066] 在图3中,虚线框所包括的起始位置为0xE:0FB0H,长度为64字节的区域属于节头部表Section Header Table的第0项,内容为全0的系统保留字段;

[0067] 紧挨着节头部表Section Header Table上方的为字符串表,字符串表的起始位置为0xE:0E42H,长度为360字节,因此,字符串表的实际结束位置是0xE:0FA9H,即该地址存储的字符为“ranges”后的字符串结束符0X0,由于需要和节头部表Section Header Table对齐,因此,字符串表后补充6个字节的空字符来形成字符串表与节头部表中间的字节空隙,地址为0xE:0FAAH至0xE:0FAFH。

[0068] 可以理解地,一个ELF文件中,还会存在多个系统保留字段和字节空隙,图3所示仅用来进行举例说明,并不对ELF文件的结构进行任何的限定。

[0069] 由图3可以得知,节头部表中64字节的系统保留字段以及字符串表与节头部表之间6个字节的字节空隙均可以用来存储ELF文件自身的校验参照值。从而能够达到隐藏校验参照值这一敏感信息,而且还不会影响ELF文件的正常运行。

[0070] 另外,为了增加敏感信息的隐蔽性,校验参照值可以分成几段或再进行变换,分别存储在这些字节空隙和保留字段中,具体实施的方案由开发人员灵活设定

[0071] 实施例三

[0072] 结合前两个实施例,在步骤S101之前,参见图4,所述方法还包括将目标ELF文件的校验参考值写入目标ELF文件中空隙字节的过程,具体可以包括:

[0073] S100A:遍历目标ELF文件,获取目标ELF文件中的空隙字节;

[0074] S100B:按照预设的写入规则将目标ELF文件的校验参照值写入所述空隙字节。

[0075] 可以理解地,步骤S100A以及步骤S100B可以由目标ELF文件的生成设备执行,本发明实施例对此不作具体限定。

[0076] 结合实施例二中所述的目标ELF文件的预设空隙字节具体可以包括:目标ELF文件中各节之间的空隙以及ELF文件各段的预留保留字段;相应地,步骤S100A可以包括:

[0077] 根据目标ELF文件中的ELF Header字段信息获取目标ELF文件各段的保留字段地址;

[0078] 根据目标ELF文件中节头部表Section Header Table获取目标ELF文件中各节之间的空隙。

[0079] 在具体实施过程中,ELF Header字段信息的结构如图5所示,以节头部表为例,图5所示结构中的参数e_shoff值为节头部表Section Header Table表的起始位置,从而可以根据节头部表Section Header Table的起始位置和节头部表Section Header Table的保留字段规则获知节头部表Section Header Table保留字段地址,也就是图3中虚线框所包括的区域。可以理解地,还能够根据ELF Header字段信息中目标ELF文件的其他各段信息以

及各段中保留字段的规则获取对应的保留字段;于是,保留字段可以用来存储校验参照值。

[0080] 节头部表Section Header Table中每个表项均对应用来描述各节Segment的信息,其中,节头部表Section Header Table用于描述节Segment信息的表项结构如图6所示,其中,sh_offset指示该节的起始地址,sh_size指示该节的数据长度;那么该节的起始地址和该节的数据长度之和则为该节的结束地址;随后遍历节头部表Section Header Table中所有表项的sh_offset,将最接近该节的结束地址的sh_offset与该节的结束地址之间的差值则为目标ELF文件中两节之间的空隙;例如,以图3为例,设定前述的字符串表为ELF文件中的第34节,因此,第34节在节头部表Section Header Table中的表项具体如图7所示,从图7中可以得出,第34节的起始地址s_offset为0x E0E42H,第34节的数据长度s_size为360;所以第34节的结束地址为0x E0E42H+360=921514=0x E0FAAH;而最接近0x E0FAAH的起始地址是节头部表Section Header Table表的起始地址,即0x E0FB0H,因此,第34节和节头部表之间的空隙为0x E0FB0-0x E0FAA=6。因此,这6个字节也能够用于保存校验参照值。

[0081] 结合图4所示的将目标ELF文件的校验参考值写入目标ELF文件中空隙字节的过程,参见图8,步骤S102所述的从目标ELF文件的预设空隙字节处读取目标ELF文件的校验参照值,具体可以包括:

[0082] S1021:获取目标ELF文件的预设空隙字节地址;

[0083] S1022:从获取到的预设空隙字节地址中读取目标ELF文件的校验参照值。

[0084] 在具体实施过程中,可以在目标ELF文件运行的代码段增加用于校验的代码,在用于校验的代码中,存储校验参照值所存储的空隙字节,从而当目标ELF文件运行至用于校验的代码时,获取到空隙字节的地址,并从获取到的空隙字节的地址读取校验参照值。

[0085] 本实施例的技术方案能够在目标ELF文件运行时,对自身进行自校验,从而无需额外的加壳步骤和解密步骤对目标ELF文件进行校验,从而能够减少ELF文件运行时占用的存储空间。

[0086] 实施例四

[0087] 基于前述实施例相同的技术构思,参见图9、其示出了本发明实施例提供了一种文件校验装置90,该装置90可以包括:运行单元901、获取单元902、读取单元903和校验单元904;其中,

[0088] 所述运行单元901,用于运行目标ELF文件;

[0089] 所述获取单元902,用于所述运行单元901运行目标ELF文件时,根据预设的校验算法获取所述目标ELF文件的实际校验值;

[0090] 所述读取单元903,用于所述运行单元901运行目标ELF文件过程中,从所述目标ELF文件的预设空隙字节处读取所述目标ELF文件的校验参照值;

[0091] 所述校验单元904,用于将所述实际校验值与所述校验参照值进行比较;以及,

[0092] 当所述实际校验值与所述校验参照值一致时,确定校验成功;以及,

[0093] 当所述实际校验值与所述校验参照值不一致时,确定校验失败。

[0094] 示例性地,所述目标ELF文件的预设空隙字节具体包括:所述目标ELF文件中各节之间的空隙以及所述目标ELF文件各段的预留保留字段。

[0095] 在上述方案中,参见图10,所述装置90还包括遍历单元905和写入单元906;其中,

[0096] 所述遍历单元905,用于遍历所述目标ELF文件,获取所述目标ELF文件中的空隙字节;

[0097] 所述写入单元906,用于按照预设的写入规则将所述目标ELF文件的校验参照值写入所述遍历单元905获取到的空隙字节。

[0098] 在上述方案中,所述获取单元902,具体用于:

[0099] 根据所述目标ELF文件中的ELF Header字段信息获取所述目标ELF文件各段的保留字段地址;以及,

[0100] 根据所述目标ELF文件中节头部表Section Header Table获取所述目标ELF文件中各节之间的空隙。

[0101] 在上述方案中,所述读取单元903,具体用于:

[0102] 获取所述目标ELF文件的预设空隙字节地址;

[0103] 从获取到的预设空隙字节地址中读取所述目标ELF文件的校验参照值。

[0104] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用硬件实施例、软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器和光学存储器等)上实施的计算机程序产品的形式。

[0105] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0106] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0107] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0108] 以上,仅为本发明的较佳实施例而已,并非用于限定本发明的保护范围。

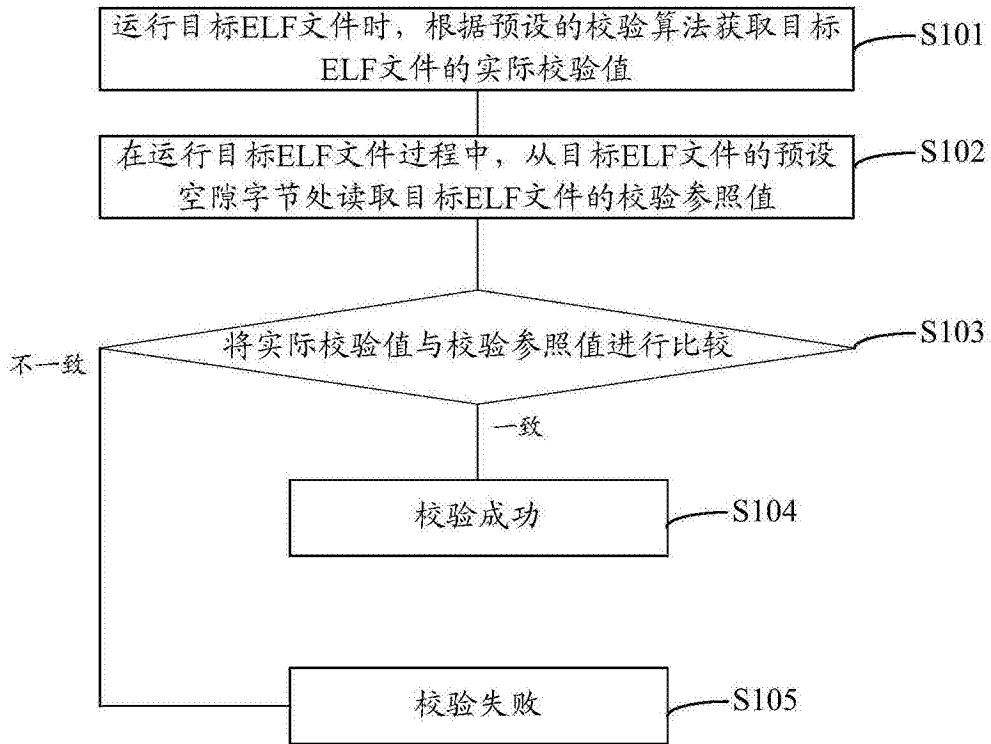


图1

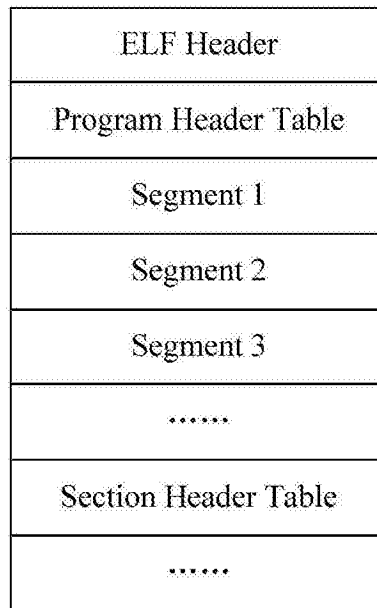


图2

E:0F30h:	00 2E 64 73 6E 61 6D 69 63 00 2E 67 6F 74 00 2E	..dynamic..got..
E:0F40h:	67 6F 74 2E 70 6C 74 00 2E 64 61 74 61 00 2E 62	got.plt..data..b
E:0F50h:	73 73 00 2E 63 6F 6D 6D 65 6E 74 00 2E 64 65 62	ss..comment..deb
E:0F60h:	75 67 5F 61 72 61 6E 67 65 73 00 2E 64 65 62 75	ug_aranges..debu
E:0F70h:	67 5F 69 6E 66 6F 00 2E 64 65 62 75 67 5F 61 62	g_info..debug_ab
E:0F80h:	62 72 65 76 00 2E 64 65 62 75 67 5F 6C 69 6E 65	brev..debug_line
E:0F90h:	00 2E 64 65 62 75 67 5F 73 74 72 00 2E 64 65 62	..debug_str..deb
E:0FA0h:	75 67 5F 72 61 6E 67 65 73 00 00 00 00 00 00	ug_ranges.(.7.7.7.)
E:0FB0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E:0FC0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
E:0FD0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
E:0FE0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
E:0FF0h:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
E:1000h:	3E 02 40 00 00 00 00 00 3E 02 00 00 00 00 00	S.@.....S.....
E:1010h:	1C 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E:1020h:	01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E:1030h:	23 00 00 00 07 00 00 00 02 00 00 00 00 00 00	*.....
E:1040h:	54 02 40 00 00 00 00 00 54 02 00 00 00 00 00	T.@.....T.....
E:1050h:	20 00 00 00 00 00 00 00 00 00 00 00 00 00 00

图3

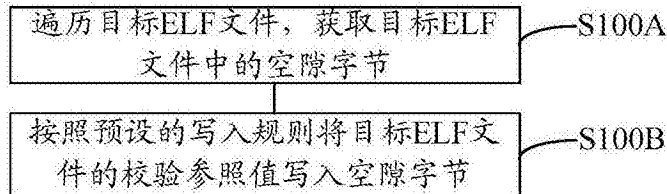


图4

```

#define EI_NIDENT 16

typedef struct {
    unsigned char  e_ident[EI_NIDENT];
    Elf32_Half    e_type;
    Elf32_Half    e_machine;
    Elf32_Word    e_version;
    Elf32_Addr    e_entry;
    Elf32_Off     e_phoff;
    Elf32_Off     e_shoff;
    Elf32_Word    e_flags;
    Elf32_Half    e_ensize;
    Elf32_Half    e_phentsize;
    Elf32_Half    e_phnum;
    Elf32_Half    e_shentsize;
    Elf32_Half    e_shnum;
    Elf32_Half    e_shstrndx;
} Elf32_Ehdr;
  
```

图5

```

typedef struct {
    Elf32_Word    sh_name;
    Elf32_Word    sh_type;
    Elf32_Word    sh_flags;
    Elf32_Addr    sh_addr;
    Elf32_Off     sh_offset;
    Elf32_Word    sh_size;
    Elf32_Word    sh_link;
    Elf32_Word    sh_info;
    Elf32_Word    sh_addralign;
    Elf32_Word    sh_entsize;
} Elf32_Shdr;
    
```

图6

struct section_table_entry64_t section_table_element[34]	
> struct s_name64_t s_name	shstrtab
enum s_type64_e s_type	SHT_STRTAB (3)
enum s_flags64_e s_flags	SF64_None (0)
Elf64_Addr s_addr	0x0000000000000000
Elf64_Off s_offset	00E42h
Elf64_Word s_size	360
Elf64_Word s_link	0
Elf64_Word s_info	0
Elf64_Word s_addralign	1
Elf64_Word s_entsize	0
char data[360]	

图7

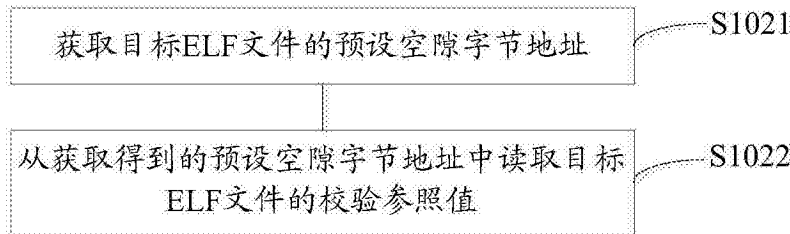


图8

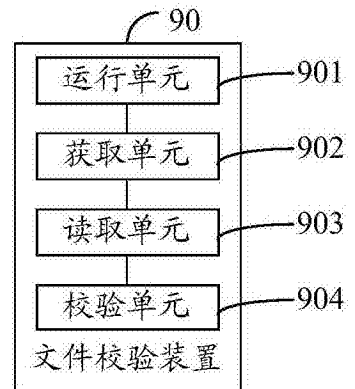


图9

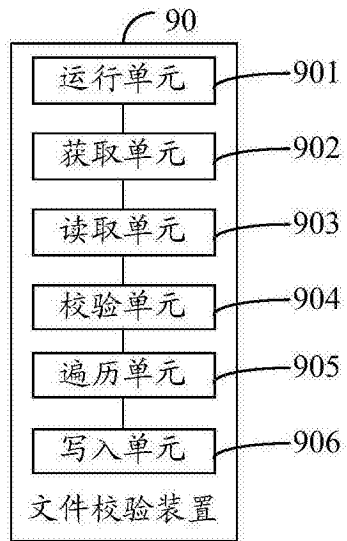


图10