



网络空间威胁对抗与防御技术研讨会  
暨 第九届安天网络安全冬训营

亂雲飛渡

资源代价与安全算力

# 量时度力

——威胁检测引擎的优化与算力支撑

安天 | 基础引擎中心

# CONTENTS

## 目 录

01

### 威胁检测引擎简介

威胁检测技术与与时俱进；引擎支撑高级威胁对抗

---

02

### 引擎的算力模型

计算场景对引擎的约束

---

03

### 引擎在算力上的优化

检测效率的优化；与硬件相结合

---

04

### 引擎结合专用芯片展望

专用芯片支撑算力

---



网络空间威胁对抗与防御技术研讨会  
暨 第九届安天网络安全冬训营

安天 | 智者安天下

# 01

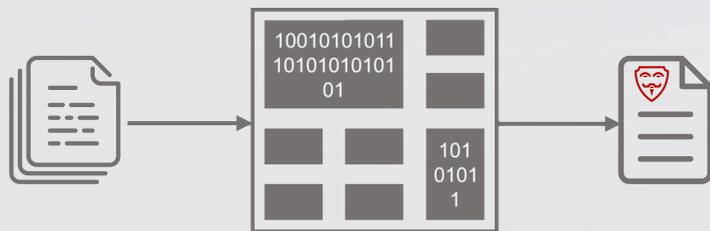
## 威胁检测引擎简介

威胁检测技术与时俱进；引擎支撑高级威胁对抗

# 什么是威胁检测引擎?

## • 威胁检测引擎的定义

一组依靠可维护规则的数据结构 通过接口调用能够对输入对象进行病毒检测处理的程序模块的统称。就像汽车的发动机是汽车的核心动力来源一样，威胁检测引擎为威胁检测产品提供着核心的鉴定能力，只要将待检测对象传入引擎，引擎即能输出对该对象的鉴定结果



## • 威胁检测引擎的应用场景

威胁检测引擎不仅可以支持传统主机反病毒产品，还可以应用到各种终端、软件、应用、设备、服务以及新兴场景中。



### 终端

移动设备、自助终端、POS机、工作站、物联网终端等



### 设备

交换机、路由器、防火墙、UTM和安全网关等



### 软件

各类主机、服务器和存储系统安全软件等



### 服务

数据中心服务和运营商级数据处理



### 应用

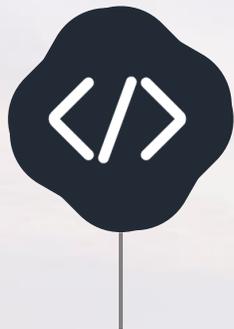
邮件服务、Web服务和代理服务等等



### 新兴场景

工业互联网、物联网、大数据、云和5G 等

# 威胁检测引擎的三高技术门槛



## 高技术难度

依赖先进科学的体系架构。面对日渐规模化、工程化、产业化的攻击者，只有以**先进的检测技术**与之抗衡，才能为网络安全保驾护航



## 高开发量

依赖大规模的分析维护团队和足够强大的计算能力。全球每天新产生的恶意代码接近百万，面对如此海量的数据，只有足够**规模的团队**和算力才能够支撑这项工作



## 高维护量

依赖长时间样本和经验的积累。与恶意代码进行对抗一方面要有对样本的广泛**捕获**和积累，以此作为分析的基础；另一方面要有**经验丰富的**工程师团队针对其进行分析与作业，以此获得检测能力

# 持续演进的安天威胁检测引擎



# 下一代威胁检测引擎的定义

## • 下一代威胁检测引擎

下一代威胁检测引擎是一组可通过接口调用，实现对输入的威胁对象进行**识别、鉴定、拆解和分析、输出处理后的信息和向量、供使用者进行威胁情报生产与消费、关联同源等工作**的程序模块的统称。

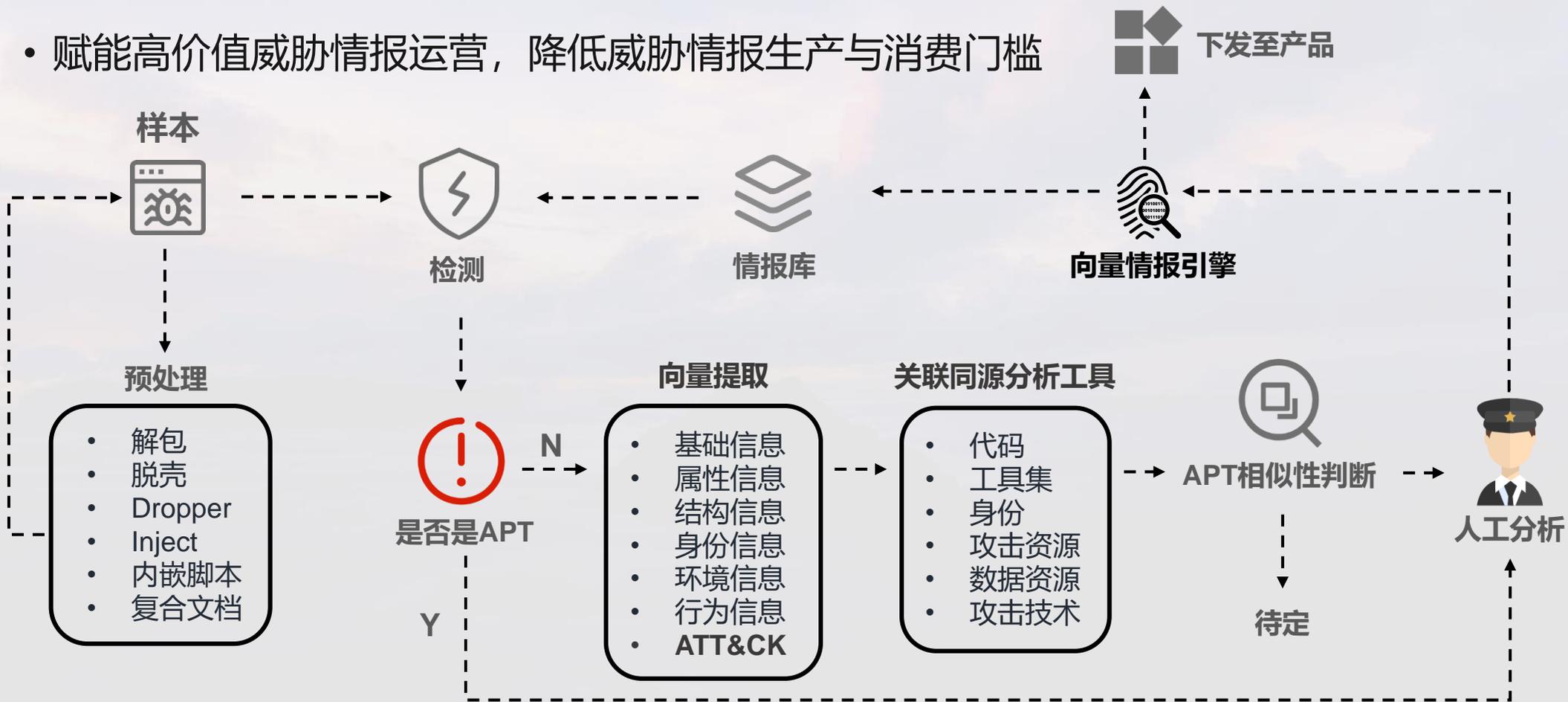
下一代威胁检测引擎是立足于在攻方可以获得防御方引擎并验证的假想下，立足于引擎本身是可以被绕过的环节来设计的。从输入对象上看，下一代引擎可以把载荷、信标、场景、人机行为作为检测向量，不再是单纯的检测，而是复合关联。把传统的结果鉴定器改为了识别器、鉴定器、拆解器、分析器的复合体，进而支撑用户场景下的**威胁追踪溯源**、技战术揭示、情报信息提取等工作。

### • 多种输入对象，多种输出结果，威胁检测多样化



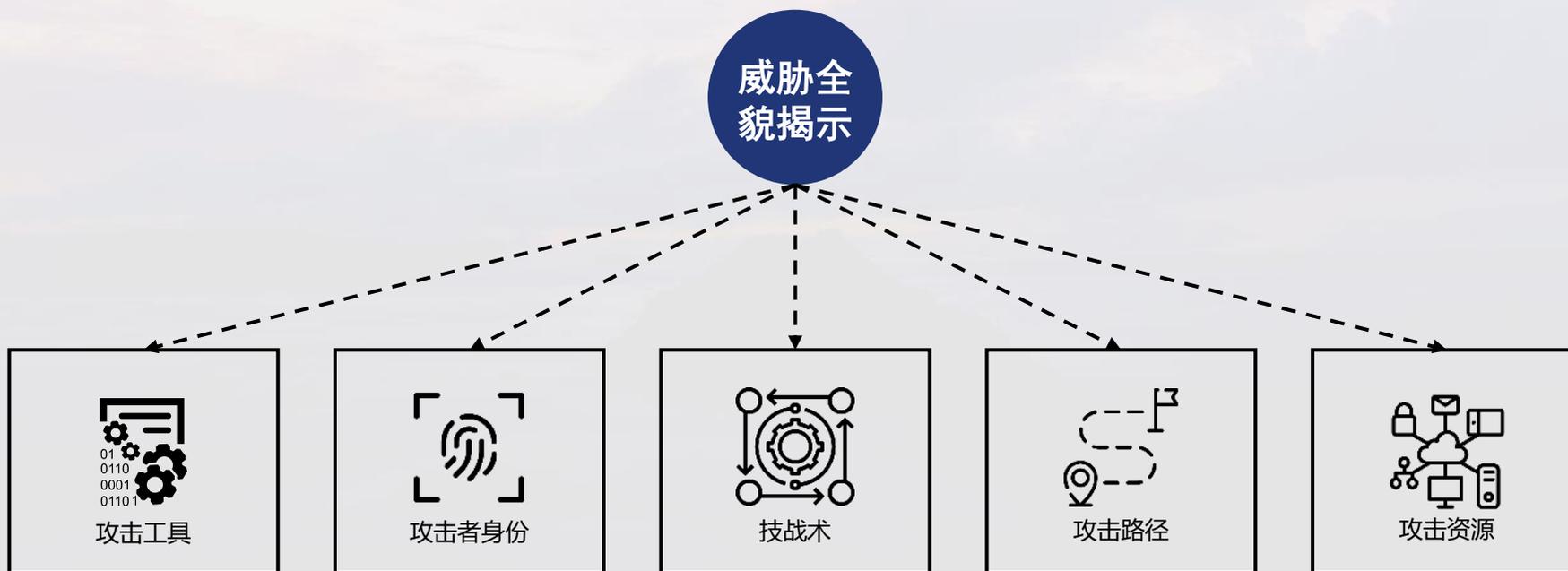
# 下一代威胁检测引擎的应用—情报的生产与消费

- 赋能高价值威胁情报运营，降低威胁情报生产与消费门槛



# 下一代威胁检测引擎的应用—赋能揭示威胁全貌

- 赋能揭示威胁全貌。基于全格式识别、深度预处理和多维向量提取等技术从样本侧揭示攻击身份、技战术、攻击资源和攻击路径等，支撑高级威胁对抗



# 下一代威胁检测引擎的应用—对ATT&CK框架的覆盖

初始访问	执行	持久化	提权	防御规避	凭证访问	发现	横向移动	收集	命令与控制	渗出	影响						
水坑攻击	利用AppleScript	利用签名的脚本代理...	利用.bash_profile和...	启动代理	利用服务器软件组件	操纵访问令牌	绕过Gatekeeper	Process Doppelgã...	操纵账户	发现账户	利用AppleScript	捕获音频	利用常用端口	自动渗出数据	删除账户权限		
利用面向公众的应用...	利用CMSTP	利用Source命令	利用辅助功能	启动守护进程	利用服务注册表权限...	填充二进制文件	修改组策略	替换进程内存	查看Bash历史	发现应用程序窗口	利用应用程序部...	自动收集	通过可移动介质通信	压缩数据	窃取数据		
利用外部远程服务	利用命令行	加入空格隐藏扩展名	操纵账户	利用Launchctl	利用Setuid和Setgid位	利用BITS服务	隐藏文件目录	进程注入	暴力破解	发现浏览器书签	利用组件对象模型(C...	收集剪贴板数据	利用连接代理	加密数据	造成恶劣影响的数据...		
添加硬件	利用HTML编译文件	利用系统中的第三方...	利用AppCert DLL(注...	添加LC_LOAD_DYLIB	修改快捷方式	利用AppInit DLL(注册...	利用启动项	绕过用户账户控制(UAC)	隐藏用户	冗余访问	凭证转储	发现域信任	利用远程服务漏洞	收集信息库数据	使用自定义C2协议	限制传输数据大小	网页内容互换攻击
通过可移动介质复制	利用组件对象模型(C...	利用Trap命令	利用AppInit DLL(注...	利用linux本地任务调...	会话发起协议(SIP)和...	利用Windows应用程...	利用Sudo命令	清除命令历史	隐藏窗口	利用Regsvcs/Regasm	获取Web浏览器凭证	发现文件和目录	执行内部鱼叉式钓鱼...	收集本地系统数据	使用自定义加密协议	通过备选协议回传	擦除磁盘内容
使用鱼叉式钓鱼附件	利用控制面板项	利用受信的开发者工具	利用Windows应用程...	利用登录项	利用启动项	绕过用户账户控制(U...	利用Sudo缓存凭证	利用CMSTP	HISTCONTROL	利用Regsvr32	获取文件中的凭证	扫描网络服务	利用登录脚本	收集网络共享驱动数据	编码数据	通过C2信道回传	擦除磁盘结构
使用鱼叉式钓鱼链接	使用动态数据交换协议...	诱导用户执行	利用认证包	利用登录脚本	利用系统固件	DLL搜索顺序劫持	利用有效账户	代码签名	映像劫持	使用Rootkit	获取注册表中的凭证	发现网络共享	利用密码哈希认证	收集可移动介质数据	混淆数据	通过其他网络介质回传	端点侧拒绝服务(DoS)
通过服务器执行鱼叉式...	通过API执行	利用Windows管理规...	利用BITS服务	利用LSASS 驱动程序	利用Systemd服务	Dylib劫持	使用Web Shell	投递后编译	阻止信标捕获	利用Rundll32	利用凭证访问漏洞	网络嗅探	利用Ticket认证	回传数据准备	前置域名	通过物理介质回传	损坏固件
入侵供应链	通过模块加载执行	利用Windows远程管...	使用Bootkit	修改现有服务	利用Windows时间服务	提示用户输入合法凭...	利用HTML编译文件	删除工具中的信标	使用脚本	强制认证	发现密码策略	利用远程桌面协议	收集电子邮件	使用域名生成算法(DGA)	定时传输	禁止系统恢复	
利用受信关系	利用主机软件漏洞	利用XSL文件执行脚本	添加浏览器扩展插件	Netsh Helper DLL	利用Trap命令	利用事件监控守护进程	利用组件对象模型(COM)劫持	间接执行命令	执行签名的二进制文...	利用Hook	发现主机接入设备	拷贝远程文件	输入捕捉	使用备用信道	网络侧拒绝服务(DoS)		
利用有效账户	利用图形用户界面(GUI)	更改默认文件关联	新建服务	利用有效账户	利用漏洞提权	额外窗口内存注入(E...	利用连接代理	安装根证书	会话发起协议(SIP)和...	欺骗用户输入凭证	发现权限组	利用远程服务	浏览器中间人攻击(MitB)	利用多跳代理	资源劫持		
	利用InstallUtil	利用组件对象模型(COM)...	启动Office应用程序	使用Web Shell	利用Windows事件订...	利用文件权限漏洞	利用控制面板项	利用InstallUtil	软件加壳	使用Kerberoasting技术	查询注册表	共享Webroot目录	捕获视频	使用多协议通信	禁用服务		
	利用Launchctl	创建账户	修改属性列表	Winlogon Helper D...	利用Hook	利用Hook	使用DShadow技术	利用Launchctl	加入空格隐藏扩展名	利用Keychain	发现远程系统	SSH劫持	使用多层加密	本地存储数据			
	利用linux本地任务调度	DLL搜索顺序劫持	端口敲门	启动守护进程	新建服务	新建服务	反混淆/解码文件或信息	LC_MAIN劫持	模板注入	LLMNR/NBT-NS投毒...	发现安全软件	污染共享内容	端口敲门	系统关机/重启			
	利用LSASS驱动程序	Dylib劫持	端口监控	启动守护进程	新建服务	新建服务	禁用安全工具	伪造	修改文件时间戳	网络嗅探	发现软件	利用系统中的第三方...	利用远程访问工具	操纵传输中的数据			
	利用Mshta	利用事件监控守护进程	利用PowerShell配置...	利用PowerShell配置...	新建服务	新建服务	DLL搜索顺序劫持	修改注册表	利用受信的开发者工具	利用Password Filter...	发现系统信息	利用Windows管理...	拷贝远程文件				
	利用PowerShell	利用外部远程服务	利用Rc.common文件	利用Rc.common文件	伪造父进程	伪造父进程	DLL旁路加载	利用Mshta	利用有效账户	收集私钥	发现系统网络配置	利用Windows远程管...	使用标准应用层协议				
	利用Regsvcs/Regasm	利用文件权限漏洞	重启应用程序	重启应用程序	路径拦截	路径拦截	按条件执行	删除网络共享连接	虚拟化/沙箱逃逸	利用Securityd内存	发现系统网络连接	发现系统所有者/用户	使用标准非应用层协议				
	利用Regsvr32	隐藏文件和目录	冗余访问	修改属性列表	端口监控	端口监控	利用漏洞规避防御	利用NTFS交换数据流...	利用Web服务	窃取Web会话Cookie	发现系统所有者/用户	发现系统服务	利用不常用端口				
	利用Rundll32	利用Hook	添加注册表运行键/启...	利用Hypervisor	利用计划任务	利用计划任务	额外窗口内存注入(EW...	混淆文件或信息	利用XSL文件执行脚本	双因子认证拦截	发现系统时间	虚拟化/沙箱逃逸	利用Web服务				
	利用计划任务	利用Hypervisor	利用计划任务	利用计划任务	利用PowerShell配置...	利用PowerShell配置...	修改文件和目录权限	伪造父进程	删除文件	修改属性列表	发现系统时间	虚拟化/沙箱逃逸					
	使用脚本	映像劫持	利用屏幕保护程序	利用屏幕保护程序	利用计划任务	利用计划任务	删除文件	修改属性列表	文件系统逻辑偏移	端口敲门							
	利用windows服务	利用内核模块和扩展	利用SSP DLL(注册表...	利用SSP DLL(注册表...													
	利用签名的二进制文...																

覆盖率达63%  
超过3000个维度的向量  
平均单样本提取向量500条

- 不相关
- 无效 (未覆盖)
- 有效
  - 可防御/可拦截
  - 可检测/可记录
  - 可降低机会
  - 可输出知识

# 下一代威胁检测引擎的应用—追踪溯源

- 应用于APT关联分析和同源分析，基于提取的多维向量为关联分析与同源分析提供更多维度更细粒度的数据支撑

T.I.Data 威胁情报综合分析平台

IP、域名、URL、HASH(MD5/SHA1/SHA256)、邮箱、字符串

黑白

## 威胁判定



Trojan/Win32.Reconyc

## 关联分析

### 同源分析

### 静态信息

### 行为分析

### 情报信息

f1799d11b34685aa20917...

3347e0c5df6fd596627870...

6983f7001de10f4d19f...

aa4c1e34950343e16dd78...

39823d50301e6a3188ba5...

5409eb407a3c7772a6436...

1768941c78b6b89aa834...

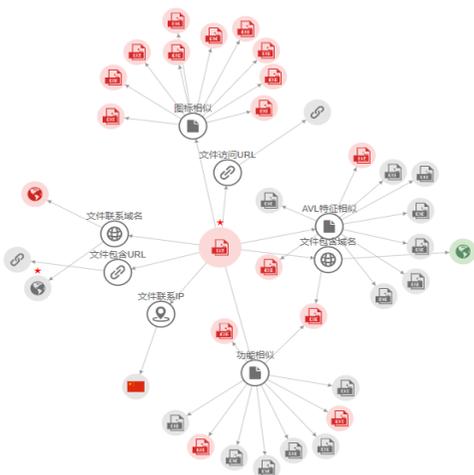
F36FAF22c22791035928...

32852E956442390E6AF3...

88DE1229D047F0AF6540...

查询历史

保存模型 导出数据 筛选关联项 显示图标



白象组织攻击样本关联分析



f1799d11b34685aa209171b0a4b89

关联 同源 收藏

判定

标签

Trojan/Win32.Reconyc

CVE-2012-4792 漏洞利用 CVE-2012-0422

CVE-2012-0158 鱼叉式网络钓鱼 APT 钓鱼 白象

邮件协议 CVE-2015-1641 CVE-2014-4114

md5

f1799d11b34685aa209171b0a4b89

sha1

98bb860344241edb96054b0f4eb4a1cc63c91

sha256

e179f03dd608b090bec933fa62d3714b6deda6c1629eect

多引擎分析

5/167

文件大小

108.413B

首发时间

2010-11-26 09:20:18

更新时间

2021-12-03 14:37:41

威胁等级

★★★★★

相关事件

白象一代活动

OPERATION HANGOVER Unveiling an Indian Cyberattack Ir

收起

关联

文件联系域名

2

功能相似

100

文件联系IP

1

文件包含URL

1

文件包含域名

1

AVL特征相似

100

文件访问URL

1

图标相似

100



网络空间威胁对抗与防御技术研讨会  
暨 第九届安天网络安全冬训营

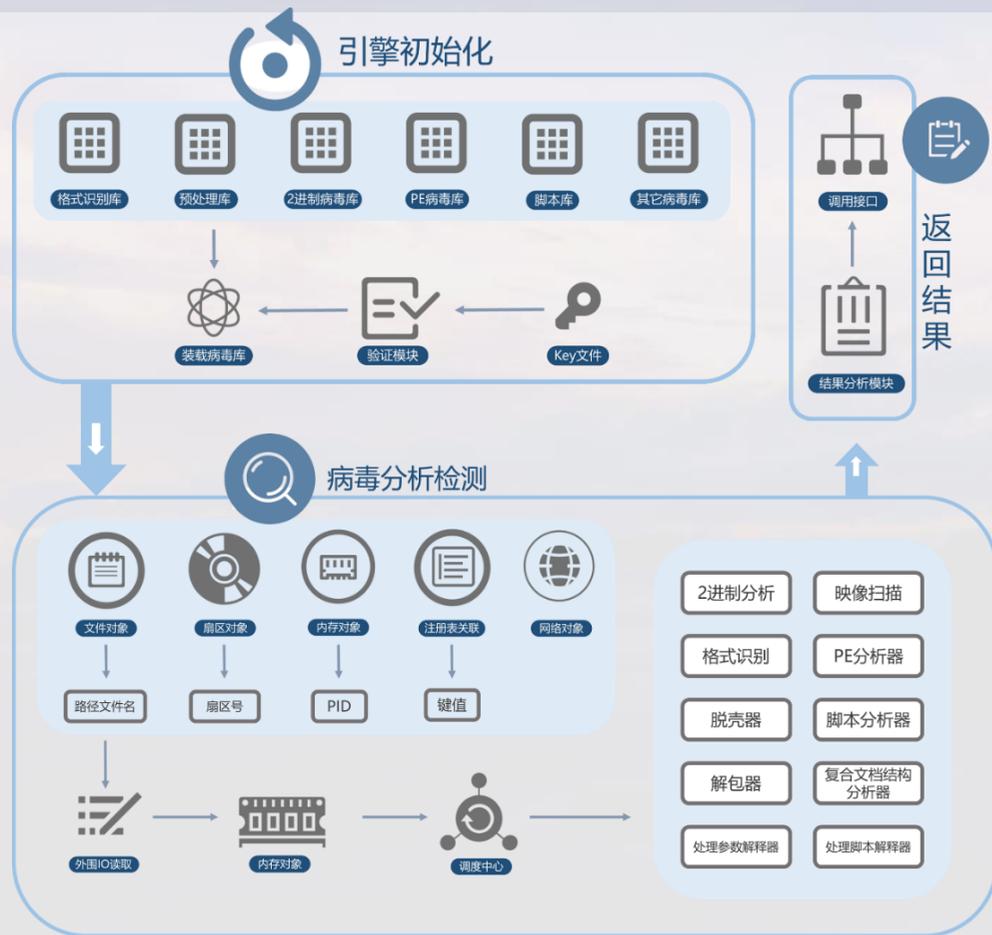
安天 | 智者安天下

# 02

## 引擎的算力模型

计算场景对引擎的约束

# 引擎的工作原理

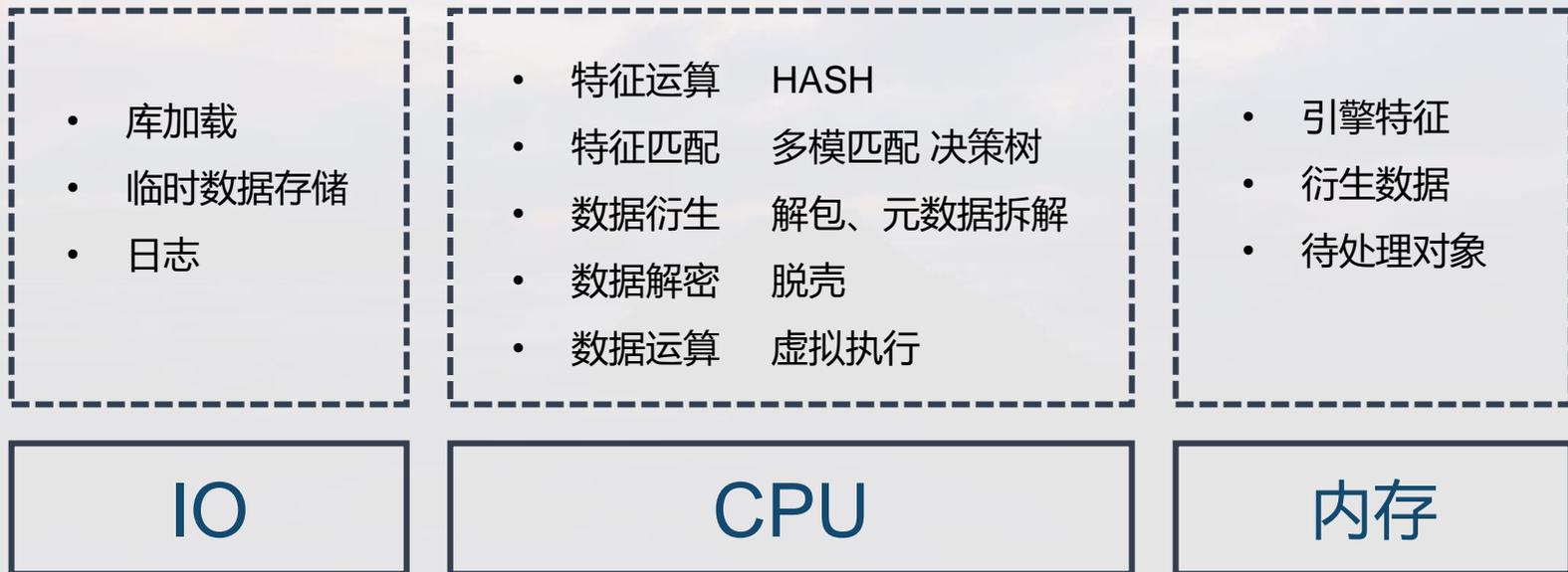


引擎的工作主要分为三个阶段：

- 一是初始化阶段，引擎会加载格式识别、预处理、二进制病毒、PE病毒、脚本病毒.....等特征库到**内存**中
- 二是分析阶段，引擎会对内存中的待检测对象进行检测分析**计算**，包括格式识别、二进制分析、解包、脱壳、复合文档分析等等动作
- 三是通过结果分析模块将检测结果输出给用户

# 引擎的算力模型

- 引擎检测的本质是依托模式识别和特征工程来实现的，这是在恶意代码时代所形成的经验。近些年来，引擎检测的对象从原来的只对部分对象进行识别检测，逐渐扩展到全量的识别与检测，这个过程是需要更大的算力基础设施去支撑的。



# 预处理对算力的消耗

<b>安天AVLSDK威胁检测引擎</b> 预处理能力参数 (统计时间： 2021年12月)	格式识别 (300种)	支持识别：可执行文件、包裹、文档、媒体文件、图片文件、软件关联格式、脚本、文本格式、其它格式等九大类格式
	解包 (58种)	压缩包 (28种)：rar, rar5, zip, tar, 7z, gzip, bzip, arj, cab, chm, iso, udf, lzh, z, zlib, xz, wim, cpio, dmg, hfs, gpt, xar, ace, alz, crx, egg, pyz, rdb
		自解压包 (6种)：zlib_sfx, rar_sfx, rar5_sfx, zip_sfx, cab_sfx, 7z_sfx
		安装包 (18种)：rpm, deb, nsis, wise, inno, installshield, setupfactory, instyler, ghostinstall, setup2go, smartinstall, vise, installcreator, tarma, autoit, quickbatch, douyou, yinginstall
	其它包裹 (6种)：Microsoft Script Encoder, binhex, mso, uue, xxe, yEnc	
	脱壳 (31种)	upx, pcompact, aspack, upack, mpress, pepatch, nspack, fsg, neolite, petite, mew, packman, kbys, pepack, rlpack, simplepack, npack, expressor, pex, beroxepack, hmimys, exe32pack, xpack, asprotect, asdpack, bambam, wwpack, depack, mkfpack, dxpack, ahpack
复合文档拆分能力 (12种)	office (7种)：rtf, doc, ppt, xls, docx, pptx, xlsx	
	adobe (2种)：pdf, swf	
	邮件 (3种)：eml, msg, tnef	

# 预处理对算力的消耗

- CPU 单核性能测试 Gzip

源文件: 16Mb

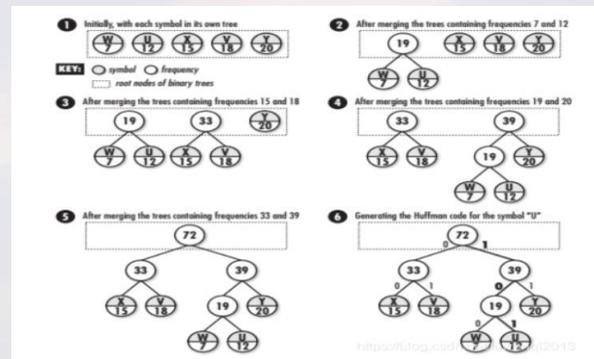
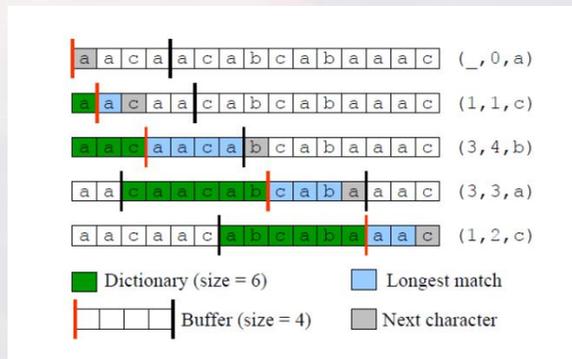
压缩后: 1600Kb

每秒解压108次

每秒可解压172.8Mb

Intel(R)Xeon(R)  
E5-2620 v4  
@ 2.10GHz

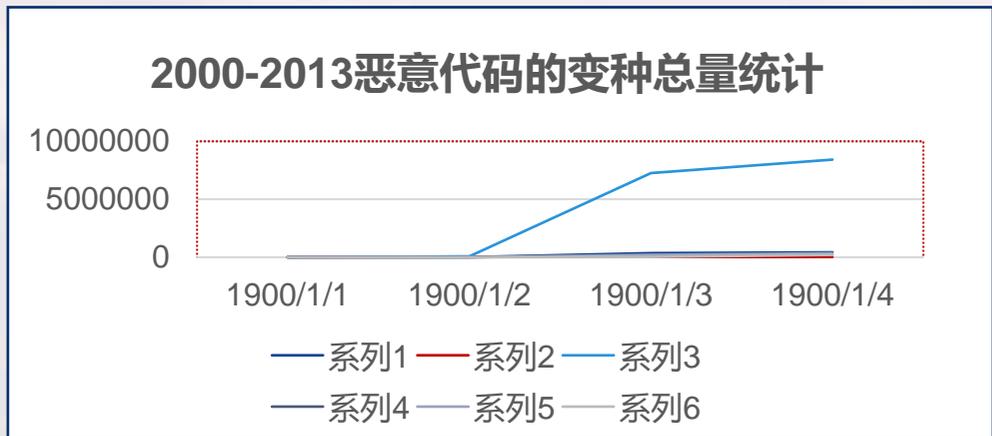
想要达到**千兆级**的性能, 大约需要**6**个核



CPU

# 算力问题的原因—数量更庞大的威胁

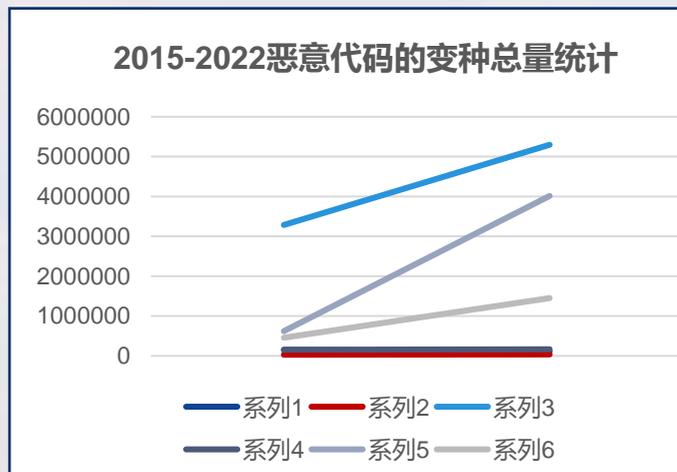
- 数量更加庞大的恶意代码需要更多的检测规则



分类/日期	2000/10/24	2006/11/10	2012/11/27	2013/11/04
Worm	512	8109	354049	435247
Virus	21006	27760	29940	30060
Trojan	3066	84811	7262094	8423751
HackTool	260	4968	217502	301076
Spyware	37	4899	214570	340751
RiskWare	0	88	25800	201401

来源: Kaspersky (卡巴斯基) 对应日期病毒名列表

从观测来看, 2014年开始, 由于卡斯基后台分析与同源合并能力的增强, 一些病毒家族和变种发生变化, 因此无法连续统计, 但总体种类膨胀超过**400倍**



2015/06/25	2022/01/11
149137	109359
29397	36821
3283882	5295977
153493	167666
622344	4013398
458035	1451458

# 算力问题的原因—数量更庞大的威胁

2000年安天病毒样本恶意代码变种数	
Trojan	3,066
Virus	21,006
Worm	512
HackTool	260
RiskWare	0
GrawWare	37
总计	24,881

2020年安天样本库恶意代码变种数	
Trojan	10,338,715
Virus	49,430
Worm	276,946
HackTool	385,361
RiskWare	1,055,493
GrawWare	2,640,863
总计	14,746,808

2000年安天引擎检测规则数	
总计	约6000条

2020年恶意代码检测规则数	
总计	52,684,446

2020年云检测规则数	
总计	664,870,674

安天引擎的本地检测规则数量，由2000年的不足**1万条**，增加到2020年超过**5000万条**，20年间增加了约**500倍**

## 2021安天引擎可识别文件格式规则

文件格式	规则数
PE	41,292,256
安卓	7,918,221
脚本	1,219,258
ELF	189,332
宏病毒	141,812
溢出	6,430
其他文件	191,713

## 2021安天引擎可解析格式

文件格式	种类数
软件数据	127
包裹	41
可执行格式	41
媒体	35
文档	31
图片	22
文本	18
脚本	9
其他文件	7

## 核心行为标签

AdTool	Email	P2P	Spoof
AdWare	Exploit	Packed	Spy
ArcBomb	FakeAV	Porn-Dialer	SuspiciousPacker
AVTool	Flooder	Porn-Downloader	Tool
Backdoor	FraudTool	Porn-Tool	VirTool
BadJoke	GameThief	Proxy	WebToolbar
Banker	Garbage	PSW	Modifier
Clicker	HackTool	PSWTool	AutoRun
Client-IRC	Hoax	Ransom	Injector
Client-P2P	IM	RemoteAdmin	Phishing
Client-SMTP	IRC	RiskTool	Toolbar
Constructor	Joke	Rootkit	KeyLogger
CrackTool	Mailfinder	Server-FTP	Filecoder
DDoS	Monitor	Server-Proxy	AdDisplay
Dialer	MultiPacked	Server-Telnet	LockScreen
DoS	Net	Server-Web	Spammer
Downloader	NetTool	SMS	HLLW
Dropper	Notifier	SpamTool	Sniffer
Bundler	Rogue		

# 算力问题的原因—手段更复杂的威胁

- 更加复杂的攻击方式使得检测机制也变得更加复杂

## 1. 原文件—>解包

```
{
  "description": "fakedir/ba1dfeecb049e654d5888323f0af853b=>quadract.dll"
},
{
  "md5": "ba1dfeecb049e654d5888323f0af853b",
  "vec": {
    "MB006771.R01": [{"VecContent": "zip"}],
    "MB006771.R02": [{"VecContent": "found_archive_head"}],
    "MB006772.R01": [{"VecContent": "data_may_be_obfuscated"}],
  },
  "description": "fakedir/aspack"
}
],
"name": "vector_analyse",
"output_timestamp": "Thu Jan 13 14:47:31 2022"
```

## 2. 解包后文件—>脱壳

```
"description": "fakedir/ba1dfeecb049e654d5888323f0af853b=>quadract.dll=>aspack"
},
{
  "size": 88612,
  "md5": "e288295f362fe23097d7788e601e8dba",
  "vec": {
    "MB002254.R01": [{"VecContent": "Encryption_algorithm"}],
    "MB002405.R01": [{"VecContent": "load_api_by_self"}],
    "MB006759.R12": [{"VecContent": "No digital signature"}],
    "MB002111.R01": [{"VecContent": "Anti_Kill_process_C"}],
    "MB006770.R01": [{"VecContent": "Packer_Compression/StarForce.ASPack"}]
  }
}
```

## 3. 脱壳后获取向量

```
{
  "input_timestamp": "Thu Jan 13 14:47:31 2022",
  "version": "1.1.2.1",
  "data": [
    {
      "size": 86016,
      "md5": "e288295f362fe23097d7788e601e8dba",
      "vec": {
        "MB002324.R01": [{"VecContent": "file_attribute_des"}],
        "MB002410.R01": [{"VecContent": "found_schtasks"}],
        "MB002259.R01": [{"VecContent": "ransom_special_file_ops"}],
        "MB001727.R01": [{"VecContent": "string_cmp_and_copy"}],
        "MB001727.R03": [{"VecContent": "memory_cmp_and_copy"}],
        "MV000012.R05": [{"VecContent": "get_clipboard_data"}],
        "MB002392.R01": [{"VecContent": "cmd_found_in_reg_parameter"}],
        "MV001872.R01": [{"VecContent": "resume_thread"}],
        "MV001872.R02": [{"VecContent": "create_thread"}],
        "MB006759.R12": [{"VecContent": "certificate": "no digital signature"}],
        "MV000029.R01": [{"VecContent": "enumerate_files_via_apis"}],
        "MB002203.R01": [{"VecContent": "create_file_to_system_directory"}],
        "MB000916.R01": [{"VecContent": "delete_shadow_copies"}],
        "MB002429.R01": [{"VecContent": "Bin_Crypt_RIPEMD160_Constants"}],
        "MB002374.R01": [{"VecContent": "Crypt_algorithm_Step_1"}]
      }
    }
  ]
}
```

发现用于设置计划任务的工具schtasks  
Scheduled Task/Job

# 算力问题的原因—手段更复杂的威胁

技术

获取访问凭据

持久化

横向扩散

执行

环境检测

数据回传

命令与控制

收集信息

逃避检测

提升权限

帐户发现	Account Discovery
应用程序窗口发现	Application Window Discovery
文件和目录发现	File and Directory Discovery
网络服务扫描	Network Service Scanning
网络共享发现	Network Share Discovery
外围设备发现	Peripheral Device Discovery
权限组发现	Permission Groups Discovery
过程发现	Process Discovery
查询注册表	Query Registry
远程系统发现	Remote System Discovery
安全软件发现	Security Software Discovery
系统信息发现	System Information Discovery
系统网络配置发现	System Network Configuration Discovery
系统网络连接发现	System Network Connections Discovery
系统所有者/用户发现	System Owner/User Discovery
系统服务发现	System Service Discovery
系统时间发现	System Time Discovery



网络空间威胁对抗与防御技术研讨会  
暨 第九届安天网络安全冬训营

安天 | 智者安天下

# 03

## 引擎在算力上的优化

引擎的优化；与硬件的结合

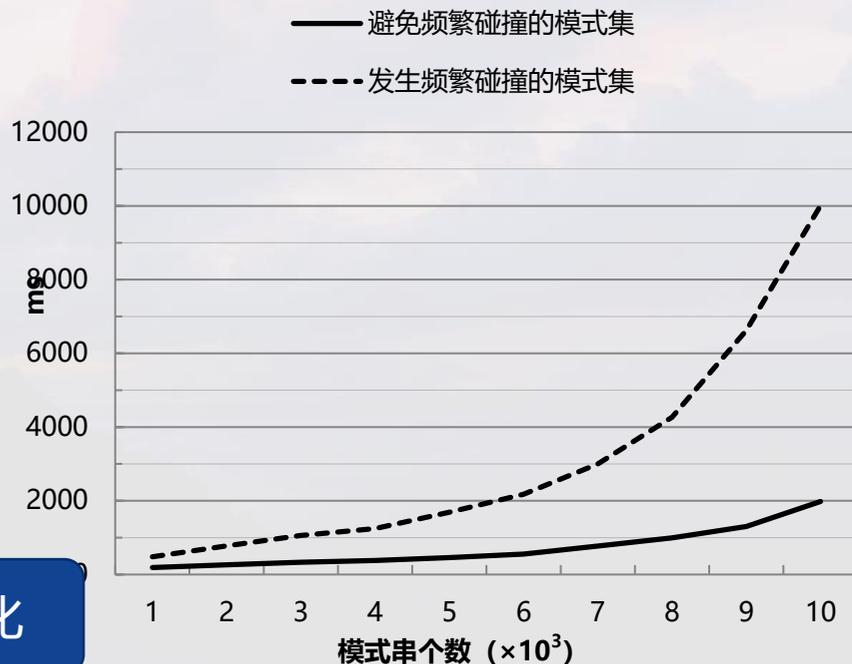
# 引擎的优化—特征维度

- 在**特征工程**方面，安天提出了网络场景下的特征优化方法，基于特征筛选技术，加速了网络场景下的检测效率，并推出了当时千兆限速检测能力的VDS，这是全球第一个可以形成千兆检测能力的网络安全设备

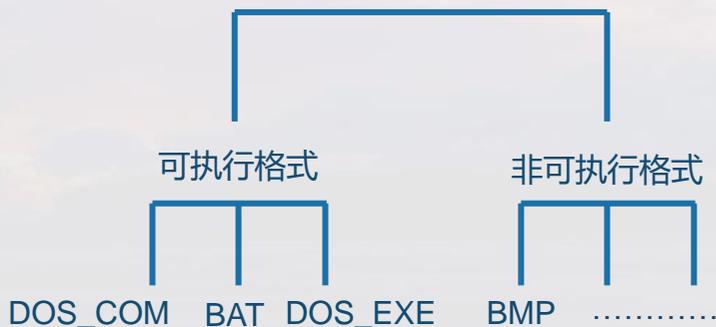
检测范围优化

算法优化

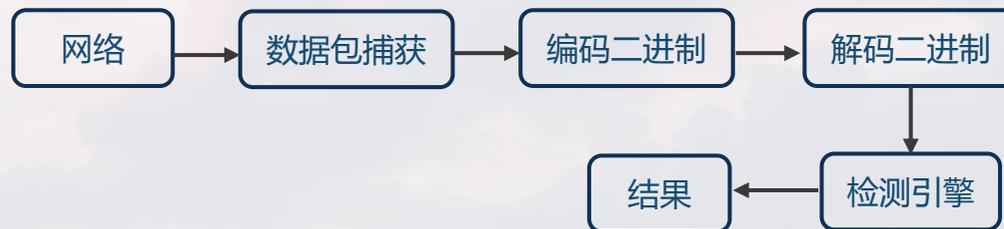
特征优化



## 引擎检测的格式分流



- 在**数据包级别**进行动态检测，减轻网络设备还原处理的压力





包处理基于指令逻辑实现，耗时较高

- ① 用**跳转表代替指令逻辑实现**，使用缓存机制减少解包时间



在数据运算、数据匹配方面，传统的方法性能受限

- ① 在虚拟运行方面，用**跳转表代替opcode指令解析**
- ② 用**状态机代替正则匹配**，提升规则匹配速度
- ③ 用**多模匹配代替单模匹配**，只需一次性匹配，无需一特征一匹配

## 1. CRC32表

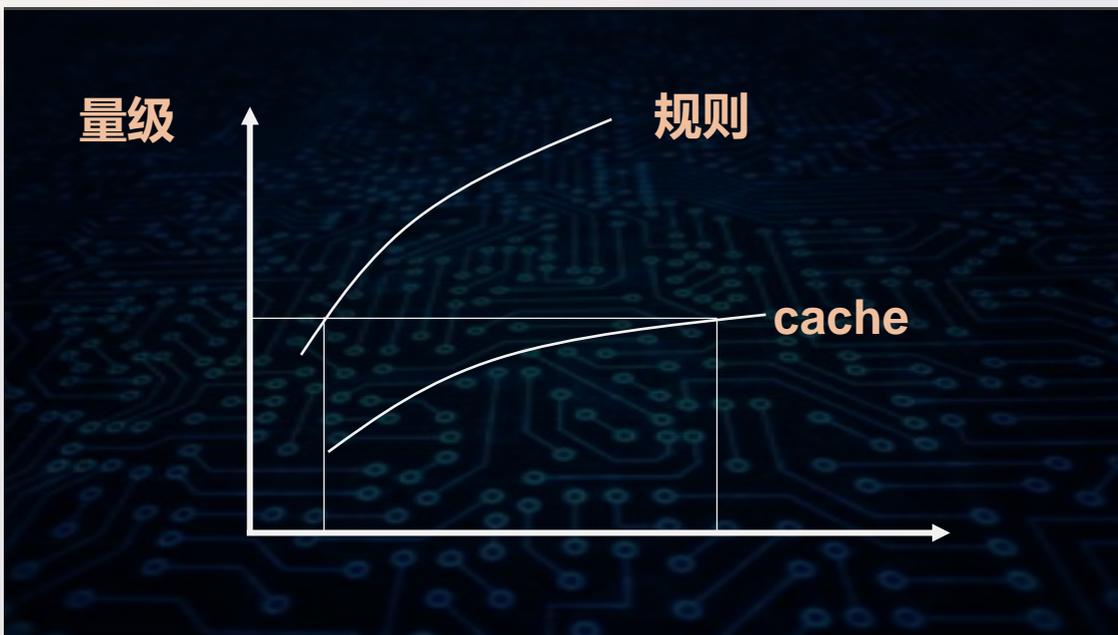
```
local const z_crc_t FAR crc_table[TBLS][256] =  
{  
  {  
    0x00000000UL, 0x77073996UL, 0xee0e612cUL, 0x990951baUL, 0x076dc419UL,  
    0x706af48fUL, 0xe963a535UL, 0x9e6495a3UL, 0x0edb8832UL, 0x79dcb804UL,  
    0xe0dd5e91UL, 0x97d2d988UL, 0x09b64c2bUL, 0x7eb17cbdUL, 0xe7b82d07UL,  
    0x90bf1a91UL, 0x1db71064UL, 0x0ab020f2UL, 0xf3b9718UL, 0x84be41deUL,  
    0x1ada047dUL, 0x6dd5de4eUL, 0xf4d4b55UL, 0x8383857UL, 0x136c985UL,  
    0x646ba8cUL, 0xf62f97aUL, 0x8a65c9ecUL, 0x14015c4FUL, 0x63066cd9UL,  
    0xfaf0f3d63UL, 0x8d080df5UL, 0x3b6e20c8UL, 0x4c69105eUL, 0xd56041e4UL,  
    0xa2677172UL, 0xc03e4d1UL, 0x4b04d47UL, 0xd20d857dUL, 0xa50ab56UL,  
    0x3555884UL, 0x42b2986UL, 0xdb0cb3ddUL, 0xacccf94UL, 0x32d86c3UL,  
    0x45df5c75UL, 0xdc60dcFUL, 0xab13d59UL, 0x26a930acUL, 0x51de003aUL,  
    0xc8d75180UL, 0xbfd06116UL, 0x21b4f4b5UL, 0x56b3c423UL, 0xcfb9599UL,  
    0xb8dca58FUL, 0x2802b9eUL, 0x5f88808UL, 0xc60c9b2UL, 0xb10e924UL,  
    0x2f1f7e37UL, 0x58684c11UL, 0x91611aaUL, 0x66662a3dUL, 0x7d6c419UL,  
    0x01db7106UL, 0x98d220bcUL, 0xef45102aUL, 0x71b18589UL, 0x00bb51FUL,  
    0x9fbfe4a5UL, 0xe8b84d33UL, 0x7807c9a2UL, 0xf0f0f93aUL, 0x9609a88eUL,  
    0x10e9818UL, 0x7f60abbUL, 0x88ed3d2dUL, 0x91646c7UL, 0xe663c01UL,  
    0x80b51f4UL, 0x1c66162UL, 0x835304dUL, 0xf22004eUL, 0x6c0998eUL,  
    0x1b01a57bUL, 0x8208f4c1UL, 0xf59fc457UL, 0x65b009c6UL, 0x12b7e95UL,  
    0x8bbbeb8eUL, 0xfcb9887cUL, 0x62dd1ddfUL, 0x15da2d49UL, 0x8cd37cf3UL,  
    0xfbd44c65UL, 0x4db26158UL, 0x3ab551ceUL, 0xa3bc0074UL, 0xd4bb30e2UL,  
    0x4ad551UL, 0x3dd9547UL, 0xa4d1c464UL, 0x3361f44bUL, 0x4369e66UL,  
    0x346e9fcUL, 0xad678846UL, 0xda60b8d0UL, 0x44042d73UL, 0x33031de5UL,  
    0xaa0a4c5FUL, 0xd0d07cc9UL, 0x5005713cUL, 0x270241aaUL, 0x9e0b1010UL,  
    0xc90c2086UL, 0x5768b525UL, 0x206f853UL, 0xb966f469UL, 0xc661e49FUL
```

## 2. 生成表

```
case the advice about DYNAMIC_CRC_TABLE is ignored) */  
if (first) {  
  first = 0;  
  
  /* make exclusive-or pattern from polynomial (0xedb88320UL) */  
  poly = 0;  
  for (n = 0; n < (int)(sizeof(p)/sizeof(unsigned char)); n++)  
    poly |= (z_crc_t)1 << (31 - p[n]);  
  
  /* generate a crc for every 8-bit value */  
  for (n = 0; n < 256; n++) {  
    c = (z_crc_t)n;  
    for (k = 0; k < 8; k++)  
      c = c & 1 ? poly ^ (c >> 1) : c >> 1;  
    crc_table[0][n] = c;  
  }  
}
```

# 探寻算力支撑—通用基础算力的限制 CPU

- 规则数量的膨胀带来存储资源的压力
- 分支众多，预处理深度加深带来计算处理资源的压力
- 热数据集庞大，局部性差异导致cache失效的性能恶化



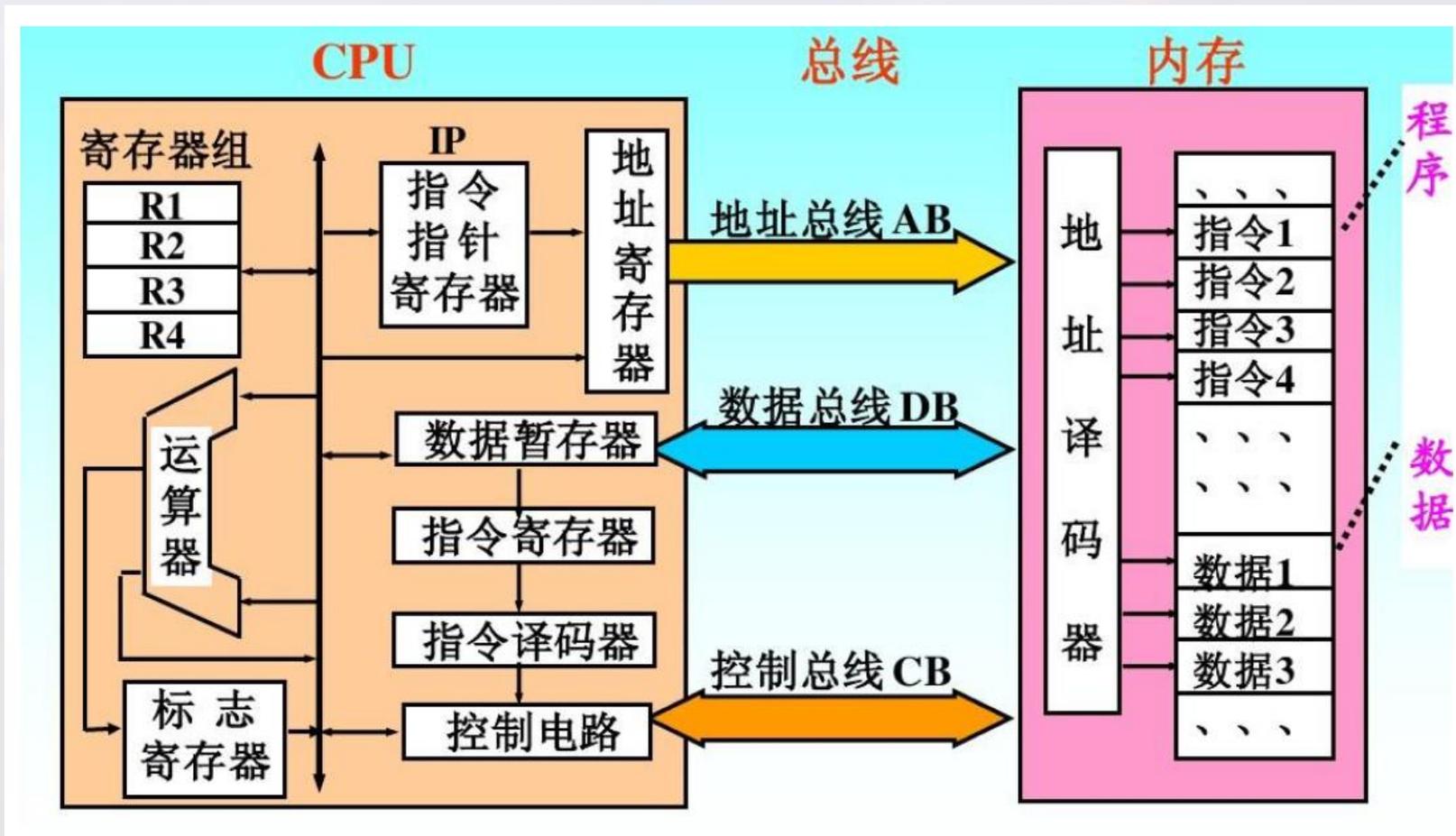
2022年CPU缓存

L1 4MB

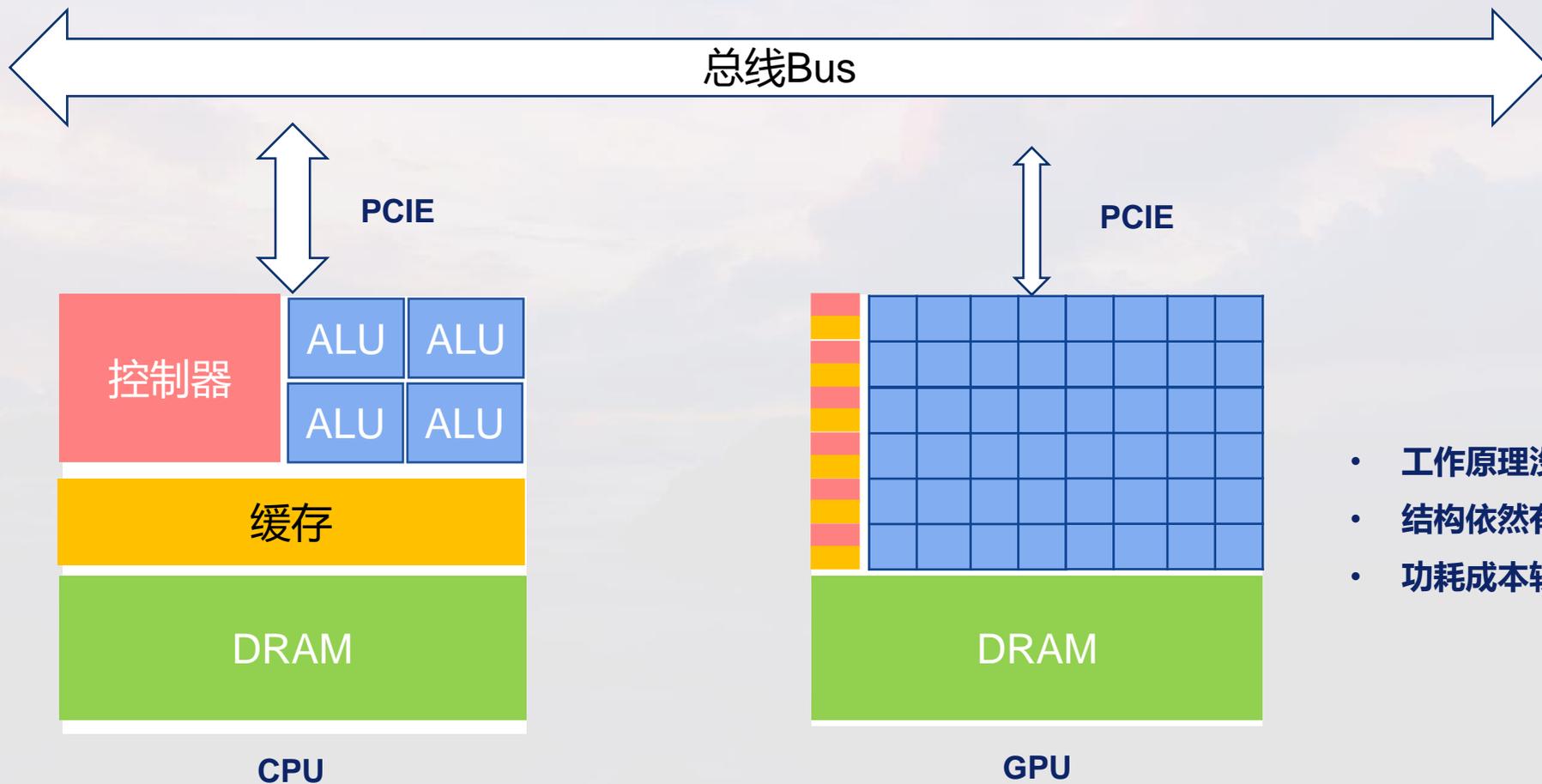
L2 32MB

L3 256MB

# 探寻算力支撑—通用基础算力的限制 CPU



# 探寻算力支撑—通用基础算力的限制 GPU



- 工作原理没有改变
- 结构依然有瓶颈
- 功耗成本较高



网络空间威胁对抗与防御技术研讨会  
暨 第九届安天网络安全冬训营

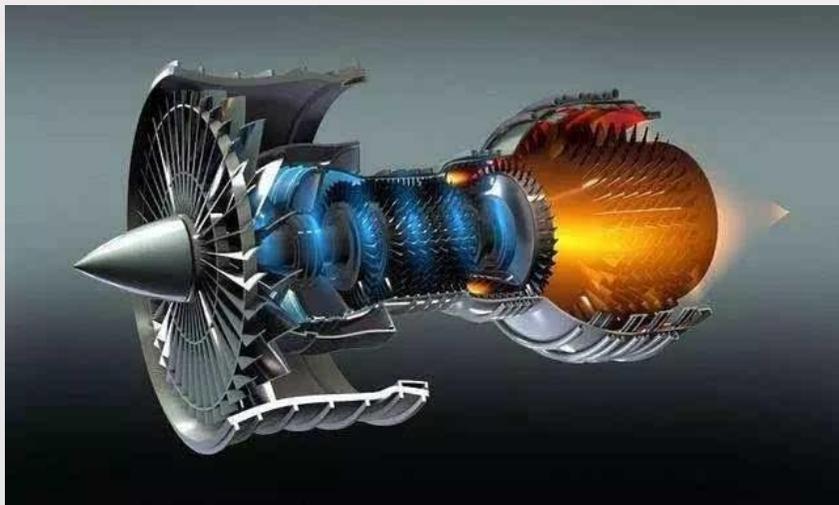
安天 | 智者安天下

# 04

## 引擎结合专用芯片展望

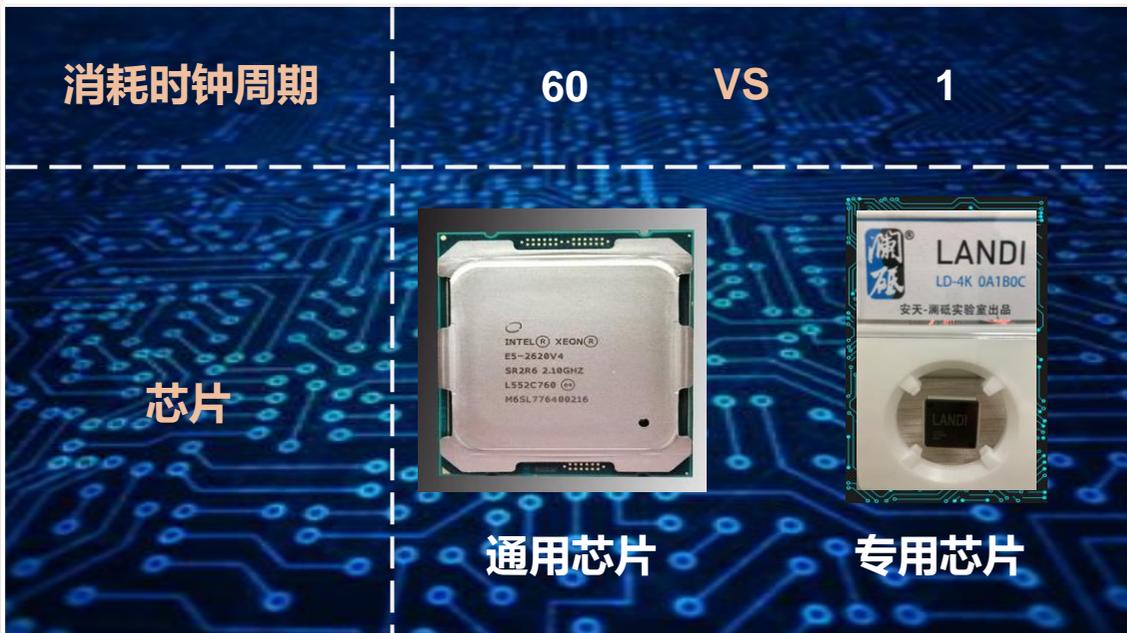
专用芯片应对算力消耗问题

- 专用芯片支撑安天引擎，为检测提供更多安全算力



# 通用 vs. 专用——性能

- 在memcmp主循环中cmp、sub指令执行时比较消耗时钟周期
- 一次循环消耗60-80个时钟周期



一次循环中的消耗时钟周期对比

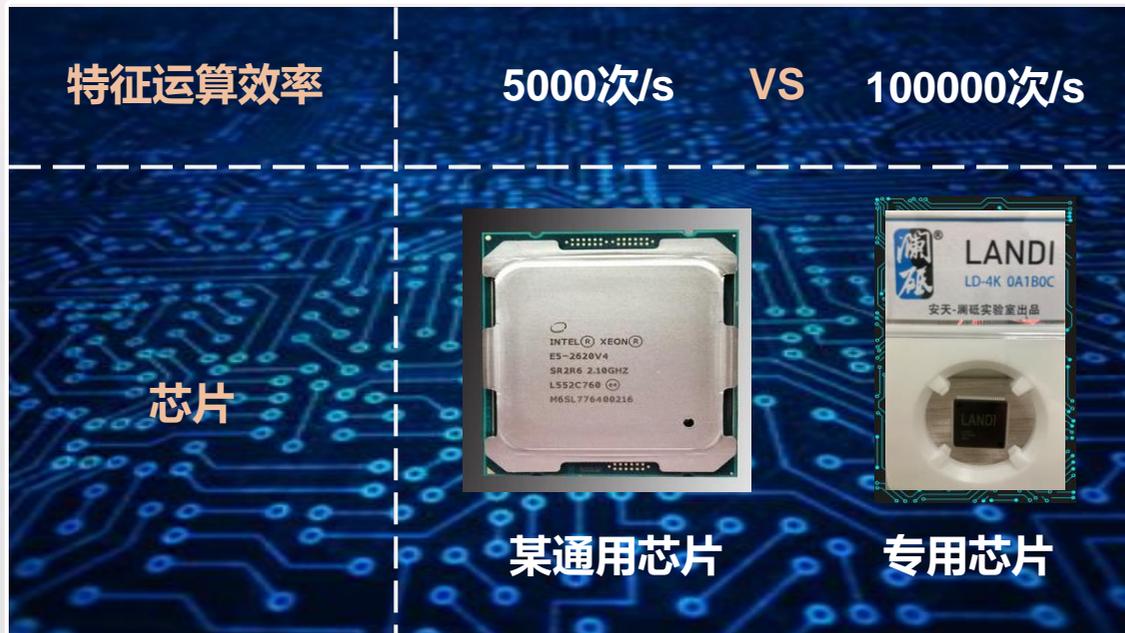
```

.text:10189146
.text:10189146 loc_10189146:                                ; CODE XREF: memcmp+24↓j
.text:10189146      mov     eax, [edx]
.text:10189148      cmp     eax, [esi]
.text:1018914A      jnz    short loc_10189158
.text:1018914C      add     edx, edi
.text:1018914E      add     esi, edi
.text:10189150      sub     ecx, edi
.text:10189152 loc_10189152:                                ; CODE XREF: memcmp+14↑j
.text:10189152      cmp     ecx, edi
.text:10189154      jnb    short loc_10189146
.text:10189156      jmp     short loc_1018915A
.text:10189158 ; -----
.text:10189158 loc_10189158:                                ; CODE XREF: memcmp+1A↑j
.text:10189158      mov     ecx, edi
.text:1018915A loc_1018915A:                                ; CODE XREF: memcmp+26↑j
.text:1018915A      test   ecx, ecx
.text:1018915C      jz     short loc_10189171
.text:1018915E      sub     esi, edx
.text:10189160 loc_10189160:                                ; CODE XREF: memcmp+3F↓j
.text:10189160      mov     al, [edx]
.text:10189162      mov     bl, [esi+edx]
.text:10189164      cmp     al, bl
.text:10189166      jb     short loc_1018917D
.text:10189168      ja     short loc_10189178
.text:1018916A      inc     edx
.text:1018916C      sub     ecx, 1
.text:1018916E      jnz    short loc_10189160
    
```

大约20条指令的一个循环

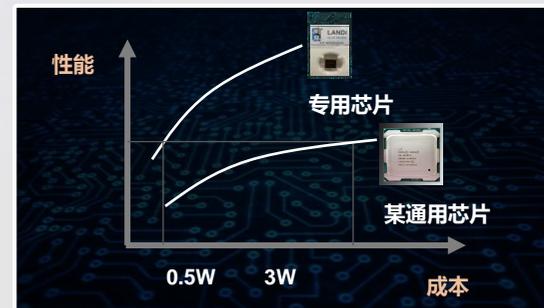
# 通用 vs. 专用——性能

- 引擎工作时在数据衍生、数据运算、特征运算和特征匹配等环节需要消耗算力
- 以特征运算环节为例，对通用和专用芯片的性能进行对比，专用约为通用的**20倍**



对平均大小为100k的文件进行特征运算

- 检测能力、效率以及算力成本的完美平衡已经成为一种现实需求，具有广阔的应用前景与市场
- 软件优化虽然在一定程度上能提升检测性能，但还是无法很好应对威胁与用户算力资源的“双重考验”
- 安全算力将会是威胁检测引擎的性能的发力点和主要发展方向之一，专用芯片具备性能和成本双重优势





网络空间威胁对抗与防御技术研讨会  
暨 第九届安天网络安全冬训营

亂雲飛渡

# 谢谢大家



安天冬训营 [wtc.antiy.cn](http://wtc.antiy.cn)